Deep Dynamic Scene Deblurring for Unconstrained Dual-Lens Cameras

M. R. Mahesh Mohan, G. K. Nithin, and A. N. Rajagopalan, *Senior Member, IEEE*

***Abstract*— Dual-lens (DL) cameras capture depth information, and hence enable several important vision applications. Most present-day DL cameras employ unconstrained settings in the two views in order to support extended functionalities. But a natural hindrance to their working is the ubiquitous motion blur encountered due to camera motion, object motion, or both. However, there exists *not* a single work for the prospective unconstrained DL cameras that addresses this problem (so called dynamic scene deblurring). Due to the unconstrained settings, degradations in the two views need not be the same, and consequently, naive deblurring approaches produce inconsistent left-right views and disrupt scene-consistent disparities. In this paper, we address this problem using Deep Learning and make three important contributions. First, we address the root cause of view-inconsistency in standard deblurring architectures using a Coherent Fusion Module. Second, we address an inherent problem in unconstrained DL deblurring that disrupts scene-consistent disparities by introducing a memory-efficient Adaptive Scale-space Approach. This signal processing formulation allows accommodation of different image-scales in the same network *without* increasing the number of parameters. Finally, we propose a module to address the Space-variant and Image-dependent nature of dynamic scene blur. We experimentally show that our proposed techniques have substantial practical merit.**

***Index Terms*— Unconstrained dual-lens camera, dynamic scene deblurring, view-consistency, multi-scale, scale-space.**

I. Introduction

MOTION blur is a prominent degradation in computer vision. The challenging problem of blind motion deblurring (BMD) deals with estimating a clean image from a motion blurred observation, without any knowledge of scene and camera motion. Since most computer vision works are designed for blur-free images and blur derails most of these tasks [5], [13], [28], BMD is a continuing research endeavour. In practice, motion blur happens due to camera motion, object motion, or both. This renders those BMD methods that are restricted to handle *only* camera-motion induced blur, ill-equipped for many practical scenarios [6], [16]. Consequently, there arises a need to seamlessly tackle blur due to camera motion and dynamic objects, so called dynamic scene deblurring. The works [13], [22], [27], [37], [39] are some notable dynamic scene deblurring methods for monocular cameras.

Apart from monocular cameras, stereo or dual-lens cameras are also widely prevalent. In traditional dual-lens cameras, typically the two cameras are identical, and they are constrained by the same camera settings and work in a synchronized fashion. The works [18], [23], [34] deal with dynamic scene deblurring in constrained dual-lens cameras. However in modern cameras, there has been a growing trend in employing unconstrained dual-lens (DL) cameras, i.e., two cameras with *same or different* focal lengths, sensor-resolutions, and exposures [15]. This flexibility supports a plethora of applications, e.g., different focal lengths and resolutions are common in today's smartphones for both narrow and wide field-of-view (FOV) functionalities; HDR imaging [1], [19], [26], low-light photography [29], and stereoscopics [20] employ different exposure times, whereas super-resolution [11], [31], visual odometry [10], [14] and segmentation [24] employ nearly-identical exposure times. More important, as in monocular cameras, motion blur is ubiquitous in DL cameras as well [15], [34], [40]. Further, the BMD methods for monocular and constrained DL are *not* applicable for unconstrained DL cameras [15]. A further challenge stems from the narrow-FOV popularized by smartphones, which amplifies the effect of camera motion and object motion, thereby exacerbating the blur. However, dynamic scene BMD for unconstrained DL cameras is still an unexplored problem.

Due to the earlier prevalence of shared camera-settings in traditional stereo cameras, most existing DL applications warrant image-pair with *identical* FOV and resolution. Similarly, stereoscopic 3D driven by the emerging demand for AR/VR (such as stereo super-resolution [11], [31], style transfer [2], [7], inpainting [30], panorama [36], etc.), warrant the input DL images to be view-consistent, i.e., the left-right view has to have coherent features as perceived by human eyes [4]. Moreover, numerous important stereo applications, such as scene-understanding, autonomous navigation and 3D reconstruction, warrant scene-consistent disparities in the DL images. For a dynamic scene BMD method for unconstrained DL cameras to serve as a potential preprocessing stage for these kind of tasks, in order to extend their scope to handle ubiquitous motion blurred observations, it has to: (I) produce view-consistent deblurred image-pair and (II) ensure scene-consistent disparities in the deblurred images.

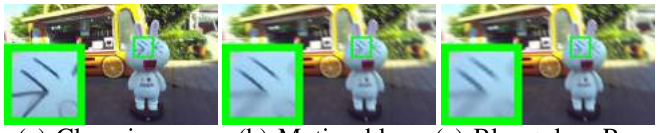
However, employing standard deblurring methods for unconstrained DL cameras *seldom* produce the desirable outcome. Motion blur and low-resolutions result in loss of high-frequency information or fine details of images

Manuscript received January 16, 2021; revised April 3, 2021; accepted April 5, 2021. Date of publication April 19, 2021; date of current version April 23, 2021. This work was supported by the Institute of Eminence (IoE) under Project SB20210832EEMHRD005001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rafal K. Mantiuk. *(Corresponding author: M. R. Mahesh Mohan.)*

The authors are with the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India (e-mail: ee14d023@ee.iitm.ac.in; ee15b047@ee.iitm.ac.in; raju@ee.iitm.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3072856, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3072856



(a) Clean image (b) Motion blur (c) Blur + low Res.

Fig. 1. Illustration of information loss: (a) Clean image, (b) Information loss due to motion blur [40], and (c) loss due to blur and resolution. ($X\frac{1}{2}$). Note the difference in degradations in (b) and (c).

(as illustrated in Fig. 1). This loss happens due to the inherent image filtering process in motion blur [32] and down-sampling [17]. Because of different exposures and resolutions in unconstrained DL set-up, the difference in the level of information-loss in the two images can be *higher* ([1], [15], [19], [26], [29]) (unlike the constrained DL case where this difference is less [40]), i.e., image in one view can have more information-loss as compared to the other view (e.g., see Fig. 4(1-2)(a-b)). Consequently, information that can be restored by a deblurring method from the two views need *not* be identical, thereby leading to view-*in*consistency. Second, it is shown in [15] that state-of-the-art deblurring methods employed for unconstrained DL-BMD can easily disrupt scene-consistent disparities.

For unconstrained DL cameras, there exists only one BMD method [15]; however, it restricts itself to *only* camera-motion induced blur. It reveals an inherent ill-posedness in unconstrained DL-BMD that disrupts scene-consistent disparities, which it addresses using a prior on camera motion. However, that prior is not *effective* when dynamic objects are present. Also, *no* attempts are made in [15] to address the problem of view-consistency. Further, it warrants an *iterative* high-dimensional optimization and hence is computationally expensive as compared to deep learning methods.

On the other hand for constrained DL-cameras, wherein both cameras share the same focal length, resolution and exposure time (with full-overlap), there exist several dynamic scene deblurring methods. These methods can be broadly categorized into two classes: Model-based optimization class and Model-agnostic deep learning class.

The model-based optimization class proceeds via a complex pipeline of segmenting different dynamic objects, estimating their corresponding motions, and deblurring and stitching different segments while suppressing possible artifacts in seams [18], [23], [34]. Due to the presence of large number of unknowns, such as segmentation masks of dynamic objects, their depths, relative motions, etc., these methods either warrant more information or restrict themselves to limited scenarios. For example, [18], [23] warrant *multiple* stereo image-pairs, whereas [34] restricts itself to motion blur primarily caused via *inplane* camera and object motions, and requires individual objects to have uniform depth. Further, due to the complex pipeline and high-dimensional optimizations involved, methods belonging to this class incur heavy computational cost.

The second class of model-agnostic methods greatly addresses the limitations of the former class, as it learns from *unrestricted* data an end-to-end mapping, that does *not* involve complex pipelines and optimizations while deblurring. However, this class is an emerging area for DL-BMD, with only one existing method [40]. As the method of [40] restricts itself to constrained set-up, the questions of ill-posedness and view-inconsistency do *not* arise [40]; but this is *not* the case for unconstrained DL-cameras wherein two cameras can have different configurations. Among other closely related works, a recent work for constrained DL [35] restricts itself to space-*in*variant Gaussian blur. Yet another issue in dynamic scene deblurring is due to its space-variant and image-dependent nature of blur [38], which is also not at all explored in the only-existing DL dynamic scene deblurring method [40].

For the first time in the literature, this paper explores the problem of dynamic scene deblurring in today's ubiquitous unconstrained DL configuration. Our work belongs to the less-explored model-agnostic deep learning class. We address the three main problems in dynamic scene deblurring for unconstrained DL cameras, namely, enforcing view-consistency, ensuring scene-consistent disparities, while addressing space-variant and image-dependent nature of blur, all in an interpretable and explainable fashion. In summary: (1). For view-consistency problem, we introduce Coherent Fusion Module with interpretable costs. Specifically, it works by fusing the unconstrained feature-pair to a single entity, which then sources a constrained feature-pair while retaining useful complementary information. (2). We show that scene-consistent disparities in the deblurred image-pair can be enforced using an Adaptive Scale-space approach. Though adaptive scale-space means directly changing the respective parameters in model-based optimization methods, there is *no* analogous flexibility in deep networks, hitherto, typically owing to their perceivance as a black-box. This we address using signal processing principles. (3). To address the space-variant, image-dependent nature (SvId) of blur, we extend the widely-used atrous spatial pyramid pooling (ASPP) [3]. Basically, it enables a neural network to produce a 'variety' of SvId receptive fields and filter-weights. Our main contributions can be summarized as:

- As a first, we address the pertinent problem of view-*in*consistency inherent in unconstrained DL deblurring, that forbids most DL applications. To this end, we propose an interpretable *Coherent Fusion Module*.
- Our work reveals an inherent issue that disrupts scene-consistent depth in generic deep learning based DL dynamic scene deblurring. To address this, we introduce a memory-efficient *Adaptive Scale-space Approach*.
- To address the space-variant and image-dependent nature of dynamic scene blur, we instil the *SvId property* in a widely-applicable deep learning module, namely atrous spatial pyramid pooling [3].
- Our proposed approach based on the above techniques achieves state-of-the-art dynamic scene BMD results for the popular unconstrained DL set-up, and we contribute the first such dataset to aid further exploration.

Rest of this paper is organized as follows: Section II brings out the reason for view-inconsistency in generic DL network, and introduces Coherent Fusion Module to address it. Section III proposes a memory-efficient Adaptive Scale-space approach to

enforce scene-consistent disparities in deblurred image-pair. Section IV discusses our SvId-ASPP in order to deal with dynamic scene blur. Section V provides various analysis and extensive comparisons, and we conclude in Section VI.

II. View-Consistency in Unconstrained DL-BMD

As compared to the constrained situation, the unconstrained scenario suffers from a higher level of difference in the information loss between the two views (due to non-uniform image-resolution and exposure). Consequently, the application of standard DL-BMD methods to this case will have more influence from such differences, leading to view-inconsistent outputs as revealed in our experimental results in Sec. V-C. To address this problem, we resort to a Deep Learning based solution. In order to motivate our solution, first we analyse the inadequacy of the only-existing deep learning based DL-BMD (albeit developed for constrained set-up) [40]. We have considered the architecture of [40] because a similar one is employed for diverse DL applications, such as style transfer [2], [7] and super-resolution [11], [31]. We first briefly review this generic DL architecture. As shown in Fig. 2, it consists of *symmetrical* networks for left-and right-view images, with both networks sharing *identical* weights (in order to *not* scale-up trainable parameters as compared to that of single-lens methods [40]). Here, the mapping from blurred images $\{\mathbf{B}^L, \mathbf{B}^R\}$ to deblurred images $\{\hat{\mathbf{L}}^L, \hat{\mathbf{L}}^R\}$ can be given as

$$\begin{aligned}\hat{\mathbf{L}}^L &= T\big(\mathbf{B}^L_{\Phi}, \mathbf{f}^L_{\Phi,fwd}, \mathbf{W} \odot \mathbf{f}^L_{\Phi,enc} + \overline{\mathbf{W}} \odot \mathbf{f}^{R\to L}_{\Phi,enc}, \mathbf{d}^L\big),\\ \hat{\mathbf{L}}^R &= T\big(\mathbf{B}^R_{\Phi'}, \mathbf{f}^R_{\Phi',fwd}, \mathbf{W}' \odot \mathbf{f}^R_{\Phi',enc} + \overline{\mathbf{W}'} \odot \mathbf{f}^{L\to R}_{\Phi',enc}, \mathbf{d}^R\big),\end{aligned} \tag{1}$$

where supersripts L/R denotes the left/right view, sets Φ and Φ' captures the resolutions and exposures of the left-right views, and $\mathbf{f}_{fwd}$ are intermediate-features of encoder which are fed-forward to decoder and $\mathbf{f}_{enc}$ is encoder-output. Bilinear mask $\mathbf{W}$ (where $\overline{\mathbf{W}} = \mathbf{1} - \mathbf{W}$) combines left-view and right-view encoder-outputs after registration (denoted by '$\to$') for view-aggregation, and $\{\mathbf{d}^L, \mathbf{d}^R\}$ are depth-features (which are outputs of a sub-network, whose inputs are disparity and intermediate features of disparity estimation network [40]).

The generic DL architecture in Fig. 2 (as employed in state-of-the-art DL-BMD network [40]) has identical mappings in the left- and right-view network. Therefore in unconstrained scenarios, there are differences in terms of resolution and/or exposure, and that leads to a higher difference in extracted features by the two networks which in turn leads to view-inconsistency. This problem is minimal for a constrained DL scenario ([40]), since the difference in extracted features will be much lesser. A similar reasoning of view-inconsistency is valid for single-image deep learning methods as well (the main difference here is that the left- and right-view networks, though can be parallelized as in Fig. 2, are decoupled, i.e., $\mathbf{W} = \mathbf{W}' = 1$ in Eq. (1)).

A. Coherent Fusion for View-Consistency

View-*in*consistency in the generic DL deblurring architecture (Fig. 2) occurs for the unconstrained case because there exist *no* avenues to reduce the difference in the extracted features by the two networks. Therefore, *one way to address the problem of view-consistency is to ensure the difference in the extracted features by the two networks to be minimal 'irrespective' of constrained or unconstrained case ($\Phi \neq \Phi'$ or $\Phi = \Phi'$)*. A careful inspection of the generic DL architecture in Fig. 2 reveals that the view-inconsistency stems from the node-pair $\{A, B\}$, where it creates imbalance in the encoder inputs and hence all feed-forward inputs to the decoder and network output (through $\mathbf{f}_{fwd}$ in Eq. (1)), which in turn creates imbalance in the decoder inputs (i.e., node-pair $\{C, D\}$).

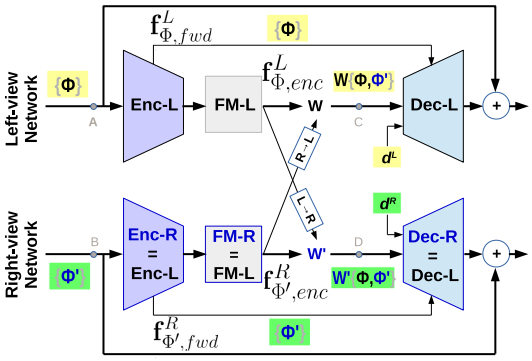


Fig. 2. Architecture of Generic DL network: For the left- and right-views (L-R), it consists of two symmetric networks with *identical* encoder (Enc), feature mapping (FM) and decoder (Dec), which produce view-consistent outputs for balanced L-R signals. However, when these identical networks process L-R imbalanced signal (due to unconstrained DL-setup), the outputs will be *view-inconsistent*.

To this end, we introduce a coherent fusion module with two self-supervision costs in the two node-pairs {A,B} and {C,D}, which enforce the following conditions: (A) The nature of output signals in the left-right views in those node-pairs are *identical*; (B) *Both* the outputs exhibit the properties of the input with *higher* information in those node-pairs. The coherent fusion at {C,D} supplements the view-aggregation of encoded features that is standard in DL networks (Fig. 2) [2], [7], [11], [31], [40], and also, enforces an additional reinforcement to balance the decoder inputs, which leads to a marginal improvement in view-consistency (Sec. V-B.2). Based on empirical observation (Sec. V-C), we select the high-resolution image as the reference (say, the right-view) since reducing resolutions leads to irrecoverable information-loss due to anti-aliasing involved. We introduce a center-view which is equidistant from the two views so that both left-right views can *symmetrically* map to it, and vice-versa. Considering the left-right view input to the module as $\{\mathbf{x}^L, \mathbf{x}^R\}$, the coherent fusion module maps $\{\mathbf{x}^L, \mathbf{x}^R\}$ to left-right view output $\{\mathbf{y}^L, \mathbf{y}^R\}$ as

$$\mathbf{y_s} = \mathbf{W} \odot \mathbf{x}^{R\to C} + \overline{\mathbf{W}} \odot \mathbf{x}^{L\to C}; \tag{2}$$

$$\mathbf{y}^L = \mathbf{W}^L \odot \mathbf{y_s}^{C\to L} + \overline{\mathbf{W}^L} \odot \mathbf{x}^L; \tag{3}$$

$$\mathbf{y}^R = \mathbf{W}^R \odot \mathbf{y_s}^{C\to R} + \overline{\mathbf{W}^R} \odot \mathbf{x}^R, \tag{4}$$

where $\mathbf{x}^{R\to C}$ registers the right-view input $\mathbf{x}^R$ to the center-view, $\odot$ is the Kronecker product, and $\{\mathbf{W}, \mathbf{W}^L, \mathbf{W}^R\}$ are image-dependent bilinear masks produced by a simple mask-generation network (as in [2], [7], [40]), i.e., $\mathbf{W}$ is

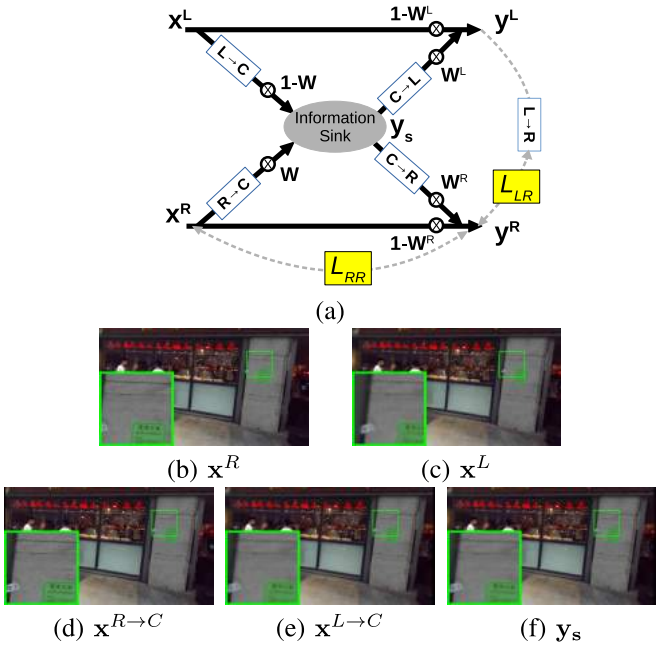


Fig. 3. (a) Coherent Fusion Module is placed in node-pairs $\{A, B\}$ and $\{C, D\}$ in the generic DL architecture (Fig. 2) to produce a balanced, yet high-feature output-pair in those nodes (from unbalanced inputs). (b-f) Visualization of registration at the center-view. (Note that Figs. (b-c) show input left-view and right-view images).

a function of the error between $\mathbf{x}^{L\to C}$ and $\mathbf{x}^{R\to C}$, where $0 \preceq \mathbf{W} \preceq 1$, $\mathbf{W}+\overline{\mathbf{W}} = \mathbf{1}$. For registration, we employ the disparity estimation method of the state-of-the-art DL deblurring network [40] and use the disparities to warp from one-view to the other-view (as in [40]). A similar approach is followed in other DL works as well [2], [7]. The main difference of our approach from [40] is that, [40] employs a warping from left/right-view to (or from) right/left-view, whereas we require warping from left/right view to (or from) the center-view and hence employ disparities multiplied by half. This is illustrated in Fig. 3(b-d) using $\mathbf{x}^{L\to C}$, $\mathbf{x}^{R\to C}$, and $\mathbf{y_s}$. In words, Eq. (2) fuses the appropriately warped left- and right-view input features to form a single center-view feature $\mathbf{y_s}$ (using $\mathbf{W}$); further, Eqs. (3)-(4) produce the output left-view (and right-view) features by merging the symmetrically-warped center-view feature and the input left-view (and right-view) features using $\mathbf{W}^L$ (and $\mathbf{W}^R$).

Also, the two self-supervision costs are

$$L_{LR} = \|\mathbf{y}^{L\to R} - \mathbf{y}^R\|_2^2 \quad \text{and} \quad L_{RR} = \|\mathbf{y}^R - \mathbf{x}^R\|_2^2. \tag{5}$$

The cost L_{LR} enforces the left- and right-view outputs to be closer to each other (enforcing Condition A), whereas the cost L_{RR} enforces right-view output feature to have higher information as that of the right-view input (enforcing condition B). (See Remark 1 in the supplementary for a justification.)

We now attempt to illustrate the working of coherent fusion module using examples. As shown in Fig. 3, the first part of the module acts as an information sink, which accumulates information from the (possibly inconsistent) input left-right views to form a *single* center-view information source. Note that the information taken from different views are image-dependent (through mask $\mathbf{W}$). This is illustrated using an example in Fig. 4, where the overall high magnitudes of mask $\mathbf{W}$ reveal that the input-view having relatively rich information (i.e., the right-view) predominantly sources the information-sink, with exceptions at occlusions or specularities where information is present only in the other view (e.g., observe the overall high-magnitude of mask of the right-view image which contains more information, except at the regions of coat behind the sailor in Fig. 4-1(a-b) or the difference in specularity behind the pillar or in the bright-window in Fig. 4-2(a-b), where right-view image does *not* have sufficient information). Finally, the center-view sources the output left-right views symmetrically, i.e., two signals with identical nature, with a provision to fill the occlusion in left-right views which is not present in the center-view (but present in the input left-right view) through masks $\{\mathbf{W}^L, \mathbf{W}^R\}$. Figures 4(d-e) provide the difference between input and output signals, which reveals a higher information flow in the left-view (as desired).

III. Consistent Depth in Unconstrained DL-BMD

Scene-consistent depth in DL images is important for diverse DL applications. In a typical stereo set-up, it requires horizontal disparities of image-features to be consistent with scene-geometry and vertical disparities to be negligible [8], [33]. Though the generic DL network with coherent fusion (in Sec. II) enforces view-consistency, it need *not* encode scene-consistent disparities in deblurred images. For the case of dynamic scenes, in general, a clean DL image-pair with scene-consistent depth is obtained when both images are captured at the *same* time-instant; otherwise, world-position(s) of dynamic object(s) in one-view need *not* be the same in the other-view which disrupts the scene-consistency. Next, we show that standard deep learning based BMD methods directly applied to unconstrained DL results in a similar issue.

A motion blurred image encodes a video sequence over its exposure time [12]; in particular, the blurred image is formed by the summation of clean frames of that video sequence [32]. Considering an unconstrained case where exposures need *not* be identical or fully-overlapping ([1], [19], [20], [29]), blurred images $\{\mathbf{B}^L, \mathbf{B}^R\}$ in the left-right views is given as [15]

$$\mathbf{B}^L = \frac{1}{t_L}\int_0^{t^L} \mathbf{L}_t^L\, dt, \qquad \mathbf{B}^R = \frac{1}{t^R - t^0}\int_{t^0}^{t^R} \mathbf{L}_t^R\, dt, \tag{6}$$

where $\{\mathbf{L}_t^L, \mathbf{L}_t^R\}$ is the clean DL image-pair obtained if, over the exposure times, the camera and scene remain at their poses at time instant t, and $[0, t^L]$ and $[t^0, t^R]$ are respectively the exposure times in the left-right views (see Fig. 5(a)). Note that the constrained DL setting is a special case of Eq. (6) with $t^0 = 0$ and $t^R = t^L$. Standard deep learning based deblurring methods works by learning a mapping from a blurred input to a clean image-pair sampled from the middle of exposure time (i.e., the centroid pivot) [27], [40]. This choice of pivot is extensively followed because for other pivots, ill-posedness exists in learning as reversing the arrow of time of the video sequence is plausible and creates the *same* blurred image (Eq. (6)), but now warrants mapping to a *different* clean image, i.e., one-to-many mapping [12]. This standard pivot is apt for

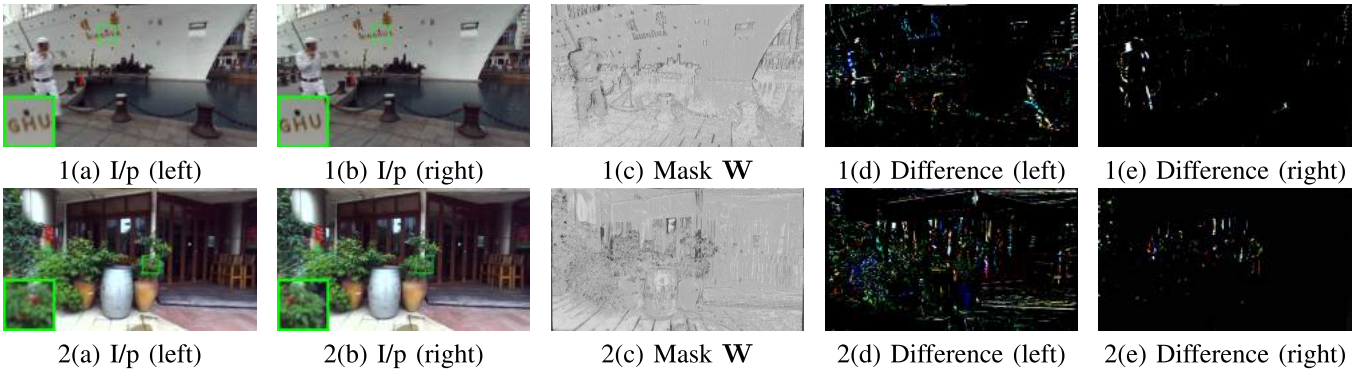
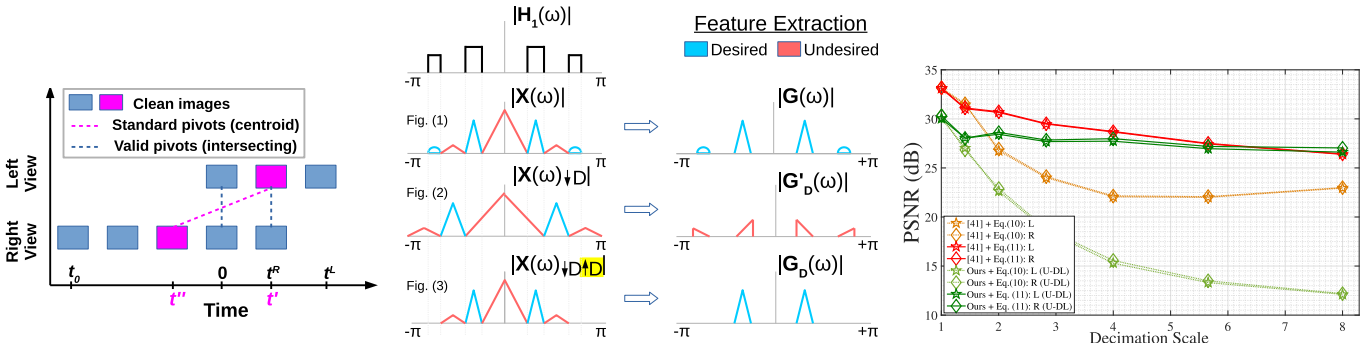


Fig. 4. Visualization of Coherent Fusion Module: (1-2) Unconstrained DL blurred images with resolution-ratio 1:2, and exposure-ratios 3:4 and 1:1, respectively (Note that input left-view/low-resolution image is shown resized to that of right-view to highlight the uneven degradations). Overall high magnitude of mask $\mathbf{W}$ reveals that the view with rich information predominantly sources the information-sink, with exceptions at occlusions or specularities where information is present only at the other view. In Figs. 1-2(b), observe the relatively rich information in right-view inputs where $\mathbf{W}$ has high magnitudes overall (Figs. 1-2(c)). Also, compare the coat behind the sailor in Figs. 1(a-b) or the specularity-difference in the pillar or bright-window in Figs. 2(a-b) where only the left-view contains the information and hence $\mathbf{W}$ magnitudes in those regions are low (Figs. 1-2(c)). Figures. 1-2(d-e) show the difference between x^L and y^L as well as x^R and y^R, which reveal a higher information flow in the left-view/low-resolution image (which has *both* blur and down-sampling) as compared to the right-view image (which has only blur). This in turn boost the deblurring performance of the left-view, unlike other methods (Fig. 9).



(a) Ambiguity in Disparity (b) Effect of Eqs. (10)-(11) for convolution (c) Implication of Adaptive Scale-space

Fig. 5. Scene-consistent Depth: (a) As centroids of blurred images need not align in unconstrained case, standard DL-BMD produces scene-*in*consistent disparities. (b) If a convolution filter is optimized for a particular signal, then its decimated version need *not* produce similar features, unless the signal is transformed to the original scale. (c) In figure, [40] + Eq. (10)/(11) denotes the original network of [40], but with introduction of Eq. (10)/(11) according to scales (L/R denotes Left/Right view). As compared to directly employing [40] to deblur different image-scales ([40] + Eq. (10)), our transformation Eq. (11) in [40], albeit a simple modification, greatly improves its performance-decay over scales. Figure also plots the performance of our network on unconstrained DL (U-DL) dataset (Case 4), which is trained using a multi-scale loss using Eq. (11). Note that our training strategy further improves the performance-decay, and the performance is sensitive to Eq. (11).

constrained DL setting, as it *automatically* results in a clean image-pair captured (or pivoted) at the same time-instant, i.e., $\{\mathbf{L}^L_{t'}, \mathbf{L}^R_{t''}\}$ where $t' = t''(= t^R/2 = t^L/2)$ [40]. However, for unconstrained case, it causes serious scene *in*consistency as the pivot t' deviates from t'' (as illustrated in Fig. 5(a)). Further, even if the pivots are chosen as some Mth and Nth fraction of exposure times, the deblurred image-pair (in general) will still exhibit scene *in*consistency, with severity increasing with the separation between the pivots $\left(M \cdot t^R \text{ and } N \cdot (t^L - t^0)\right)$. Hence, for an unconstrained set-up where timings $\{t^R, t^0, t^L\}$ *freely* vary, there does *not* exist a unique choice of pivots which produces scene-consistent disparities.

A. Adaptive Scale-Space for Scene-Consistent Depth

Scene-inconsistent disparities occur in dynamic scene unconstrained DL-BMD due to non-intersecting pivots. Therefore, *a method to address this problem in dynamic scene DL-BMD has to establish a mutual agreement between the left- and right-view images to arrive at an intersecting pivot* (as shown in Fig. 5(a)). Since single-image networks for DL-BMD operate by *independently* reusing the same network for the two views [15], [40], a mutual agreement *cannot* be established between the views. Even though the generic DL architecture promotes a signal-flow between the views, i.e., by combining *registered* encoder-output of one view with that of the other view (in node-pair $\{C, D\}$ of Fig. 2), which is indispensable for coherently adding the two encoder-outputs [2], [7], [40], this registration hinders the control on pivots. Further, the prior developed in the DL-BMD method [15] to address scene-inconsistent depth is inadequate for dynamic scenes (please see our Supplementary (Sec. S2a) for a justification).

We first provide an outline of our solution. We show that if DL-BMD is performed in lower scales (i.e., on decimated DL blurred images), the problem of depth discrepancy becomes less severe. This motivates our *adaptive* scale-space

approach for unconstrained DL deblurring. For a given DL blur input, we start deblurring from an *appropriate* lower scale where the depth discrepancy is negligible in order to produce scene-consistent results in that scale, which then progressively correct depth discrepancies in subsequent higher scales till the fine scale is reached. By 'appropriate lower scale' we mean that one network-level is shown effective for a constrained DL case [40], whereas for an unconstrained case, as depth discrepancy becomes higher more network-levels are required. In the following, we elaborate this approach in detail.

Considering the standard centroid pivot [12], [27], [40], we attempt to quantify the disparity error in unconstrained DL-BMD. Let the world coordinate of a scene-point at time-instant t' is $\mathbf{X}$; then its corresponding image-coordinates at the same pivot t' in the left- and right-views are respectively $\{\mathbf{K}^L\big(\frac{\mathbf{X}}{Z}\big), \mathbf{K}^R\big(\frac{\mathbf{X}+\mathbf{l_b}}{Z}\big)\}$, where $\mathbf{K}^L$ and $\mathbf{K}^R$ are the corresponding intrinsic camera matrices, $\mathbf{l_b}$ is the stereo baseline, and Z is the scene-depth [15]. The matrix $\mathbf{K}$ is of the form diag(f, f, 1), where f is the focal length in pixels which is proportional to the number of image rows or columns [15], [32]. Note that this case produces *zero* disparity error as the right-view sees the same world-coordinate as the left-view, displaced by the baseline (as in constrained DL-BMD).

Next, suppose that the scene-point $\mathbf{X}$ has undergone a pose-change at time-instant t'', which is modelled by rotation and translation $\mathbf{R}$ and $\mathbf{t}$ (i.e., $\mathbf{RX} + \mathbf{t}$, with corresponding depth Z'). Assume that the left and right-views have pivots at $\{t', t''\}$ (as shown in Fig. 5(a)). In this case, corresponding image-coordinates become $\{\mathbf{K}^L\big(\frac{\mathbf{X}}{Z}\big), \mathbf{K}^R\big(\frac{\mathbf{RX}+\mathbf{t}+\mathbf{l_b}}{Z'}\big)\}$. Clearly, the latter case exhibits a scene-inconsistent offset in the right-view as compared to the previous case, which is given as

$$\Delta\mathbf{x}^R = \mathbf{K}^R\left(\frac{\mathbf{X}+\mathbf{l_b}}{Z} - \frac{\mathbf{RX}+\mathbf{t}+\mathbf{l_b}}{Z'}\right), \tag{7}$$

where $\Delta\mathbf{x}^R$ is the image-coordinate discrepancy which contributes to scene-inconsistent depth. Now consider that we decimate the blurred image-pair by a factor of $D(>1)$, i.e., image resolutions are scaled-down by D and hence the focal lengths (in pixels) will be scaled by $1/D$ [8], [32]. Therefore the resultant $\Delta\mathbf{x}^R$ in Eq. (7) becomes

$$\Delta\mathbf{x}^R_D = \mathbf{D}\Delta\mathbf{x}^R, \text{ where } \mathbf{D} = \text{diag}\left\{\frac{1}{D}, \frac{1}{D}, 1\right\}. \tag{8}$$

An important insight from Eqs. (7)-(8) is that *image-coordinate discrepancies get scaled down in accordance with decimation factors*. This serves the basis for our adaptive scale-space approach (Fig. 5(b)). First, we judiciously select a decimation factor so that the disparity error is very small [8], [9]. Next, we consider the coherent deblurred image-pair from the selected scale as the reference to centroid-align the binocularly inconsistent blurred image-pair in the higher scale (via registration), which similarly produces a coherent deblurred image-pair. This process is repeated till the fine-scale. Note that our registration approach is similar to the video deblurring method [25] where a blurred frame is used as the reference to centroid-align its neighbouring blurred frames, which together produce a coherent deblurred frame. Further, employing deblurred image from a coarse scale as the reference for higher scale is standard practice in conventional deblurring methods [15], [32].

B. Memory-Efficient Adaptive Scale-Space Learning

However, existing deep learning based scale-space approach ([6], [16], [27]) in this regard has several limitations. First, as it is typically designed for a pre-determined network-levels (exactly three) it imposes an upper limit on allowable depth discrepancies, and hence *ineffective* for a large class of unconstrained DL inputs. Second, yet simply increasing the network-levels is *not* desirable as it calls for *independent* network-weights for different image-scales which escalates the memory requirement [6]. Third, as this approach uses the same network-levels *irrespective* of constrained and various unconstrained cases, it increases the computational cost (with respect to both FLOPs and processing time); to this end, we devise a training/testing strategy in Sec. V-A which *appropriately* selects the lower scale depending on the input case. Here, we address the first two limitations using signal processing principles, where we show how to optimally convert a fine-scale network to a multi-scale network by reusing the *same* weights, thereby allowing desired number of multi-levels while *not* escalating the memory requirement.

Assume that our DL-BMD network is optimally trained for the fine-scale. Since the coherent fusion (Sec. II-A) creates symmetric networks for the two views, the inference derived for a particular view is valid for the other as well. In frequency domain, the overall network-mapping for a view is given as

$$\mathbf{Y}(\omega) = \mathbf{X}(\omega) + T\big(\mathbf{X}(\omega)\big), \tag{9}$$

where $\mathbf{Y}$ is the output of the network and $\mathbf{X}$ is the respective output of the first coherent-fusion module (after node-pair $\{A, B\}$ in Fig. 2), and $T(\cdot)$ is the mapping of encoder-decoder network which involves a series of convolutions and other non-linear operations. Next, consider that the network optimized for the fine-scale (Eq. (9)) is directly employed for a lower image-scale ($D > 1$). For decimated input images, the costs of the first coherent fusion module (Eq. 5) enforces its output as a decimated version of its fine-scale output ($\mathbf{X}(\omega)$). Denoting decimation by $\downarrow D$, the network-mapping becomes

$$\mathbf{Y}'_{\mathbf{D}}(\omega) = \mathbf{X}(\omega)_{\downarrow D} + T\big(\mathbf{X}(\omega)_{\downarrow D}\big), \tag{10}$$

We claim (as in [6]) that considering the fine-scale network as a black-box for other lower-scales (i.e., Eq. (10) is *not* optimal as it maps to complimentary features. However, the following network-mapping addresses this limitation:

$$\mathbf{Y_D}(\omega) = \Big(\mathbf{X}(\omega)_{\downarrow D\uparrow D} + T\big(\mathbf{X}(\omega)_{\downarrow D\uparrow D}\big)\Big)_{\downarrow D}, \tag{11}$$

where $\uparrow D$ denotes interpolation. We assume that anti-aliasing and anti-imaging filters in decimation and interpolation [17] are ideal, but our training procedure discussed at the end of this section relaxes this. In general, $\mathbf{X}(\omega)_{\downarrow D\uparrow D} \neq \mathbf{X}(\omega)$.

Remark 2: Individual functions of DL-BMD network, such as convolution filters, bias, ReLU, decimation/interpolation, etc., optimized for fine-scale is applicable to lower scales

under the transformation of Eq. (11) (as compared to Eq, (10)). (Please see our Supplementary for a detailed justification.) We attempt to provide some intuitions using convolution. As compared to Eq. (10), basically Eq. (11) interpolates the network-input before feeding to the network, and finally decimates the network-output. In a scale-space network, the output of lower-scale network should be the decimated version of the fine-scale network-output [6], [15], [16], [27], [32]. This implies that there exists analogous frequency features in lower-scale network-output as that of fine-scale network-output (but scaled in frequency domain) [17]. As shown in Fig. 5(b1), suppose that the transformation $T(\cdot)$ at a fine-scale (in Eq. (9)) extract/map features in a particular frequency band (desired features), whereas suppresses features in the remaining frequency band (undesired features). Then to leverage the same transformation $T(\cdot)$ for lower-scales and yet retain those desired features (as required in a scale-space network), the domain of the desired features for lower scales should be *identical* to that of fine-scale. However, decimation of inputs in lower-scales (as followed in Eq. (11)) alters the domain of frequency features by expanding the frequency spectrum, and hence it extract/map complementary features instead of desired features (Fig. 5(b2)). To this end, Eq. (11) initially interpolates the decimated inputs in lower scales in order to make the domain of desired features identical for all scales (Fig. 5(b3)). Finally, Eq. (11) decimates the network-output which scales down the discrepancies in accordance with Eq. (8).

We now experimentally validate Remark 2. In line with our assumption that the network $T(\cdot)$ is optimally trained for the fine-scale, we considered the publicly-available DL-BMD network [40] (without modifying its weights) to underscore the importance of Eq. (11) over Eq. (10). We evaluate the state-of-the-art BMD [40] for different image-scales over its dataset (in Fig. 5(c): [40] + Eq. (10)). Clearly, the performance drastically drops with lower scales (validating the ineffectiveness of Eq. (10)). Next we introduce our transformation in Eq. (11) to [40] (in Fig. 5(c): [40] + Eq. (11)). Though a simple modification, our proposed approach significantly improves the earlier performance-decay of [40](Fig. 5(c)). The prevailing slight decay of performance can be possibly due to our assumption of ideal anti-aliasing and anti-imaging filter. We next discuss a training strategy we employed in our network to further improve the performance-decay. Specifically, a network trained *only* for the fine-scale need not optimize the filters to perform ideal anti-aliasing and anti-imaging. To this end, we train our network in a scale-space manner by deriving the lower-scale networks using Eq. (11) with *tied* network-weights, and using BMD costs averaged over *all* image-scales. Figure 5(c) plots the performance of our fine-scale network (ours + Eq. (11)) for different image-scales, which clearly reveals a noticeable improvement in performance-decay. This is because our training strategy demands the fine-scale network to realize effective anti-aliasing and anti-imaging through trainable filters, in order to optimize the performances in lower scales too (via the joint-cost). Also, note the higher sensitivity of the above network when we ablate Eq. (11) (in Fig. 5(c): ours + Eq. (10)).

IV. Image-Dependent, Space-Variant Deblurring

In this section, we focus on yet another issue that stems from directly employing the network-modules of the state-of-the-art DL deblurring method [40] with coherent-fusion (Sec. II-A) and adaptive scale-space (Sec. III-A). As dynamic scene blur is typically space-variant and image-dependent [38], an effective dynamic-scene deblurring network warrants a mapping that varies with spatial locations (with varying receptive fields), and that adaptively varies with different blurred images [21], [38]. Intuitively, consider a scenario of static camera, and two dynamic objects at different depths moving with the same velocity. Here, the static background exhibits no motion blur, whereas the nearer object exhibits more blur than the farther one (due to parallax [40]). Therefore, an effective deblurring network warrants an identity mapping for the background and non-identity mapping for the dynamic objects, but with a relatively larger receptive field for the nearer object. Also, positions of those dynamic objects can vary for different images, and hence the mappings need to be image-dependent. However, the filters of a generic CNN (as in [40]) are spatially invariant (with spatially-uniform receptive field), and are independent over images [21], [38]. Therefore, the only-existing DL deblurring network [40] does *not* admit a space-variant image-dependent mapping.

As discussed in [40], one key component that leads to its good performance improvement is the context module (FM in Fig. 2), which is a slightly modified version of atrous spatial pyramid pooling (ASPP) [3]. This is because the ASPP offers a good trade-off between accurate localization (small receptive field) and context assimilation (large receptive field). Note that ASPP has also been adopted for a broader set of tasks, such as semantic segmentation, object detection, visual question answering, and optical flow; however it does not support SvId mapping. Owing to the presence of both small and large receptive fields in ASPP, we extend the ASPP module to instil the SvId property in our deblurring network.

First we briefly discuss about the ASPP module [3]. The ASPP probes an input with filters at multiple sampling rates and receptive fields. This is efficiently implemented using multiple parallel atrous convolutional layers with spatially small filters (3×3) but with different sampling rates, i.e., each filter is associated with a rate $r \geq 1$, which introduces $r-1$ zeros between consecutive filter values, thereby effectively enlarging the kernel size without increasing the number of parameters or the amount of computation. Mathematically, the mapping of an input $\mathbf{x}$ in an ASPP module is given as

$$\mathbf{y} = \mathbf{x} * (\mathbf{k}_{e1} + \mathbf{k}_{e2} + \cdots + \mathbf{k}_{ep}), \tag{12}$$

where $\mathbf{k}_{ei}$ is the filter at ith branch with sampling rates ei such that $e1 = 1$ and $ep > \cdots > e2 > e1$. Clearly, even though the individual filters $\mathbf{k}_{ei}$ possess diverse receptive fields, there exists only one resultant filter realization in Eq. (12) with the receptive field as that of $\mathbf{k}_{ep}$. Also, this filter realization is identical in all spatial coordinates and is irrespective of input. Therefore, the ASPP module [3] *cannot* admit an SvId mapping, which is desirable for dynamic scene BMD.

Next, we propose a simple but effective modification to ASPP to enable the SvId property. As shown in Fig. 6,

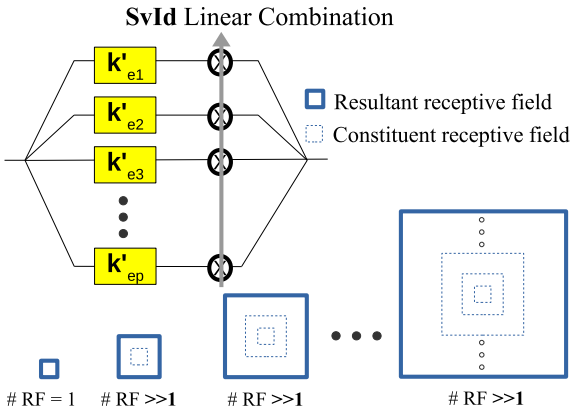


Fig. 6. Space-variant, image-dependent (SvId) atrous spatial pyramid pooling (ASPP): The ASPP [3] produces *only one* resultant filter (RF) with receptive field as that of the constituent filter with maximum field-of-view (in Fig., RF in the far-right). As this filter realization is *same* for all spatial coordinates *irrespective* of input, it does *not* admit SvId property. SvId-ASPP has the freedom to produce numerous RFs with receptive field as that of any constituent filter through SvId linear combinations of filtered outputs in individual branches.

we introduce in each parallel branch of ASPP a space-variant, input-dependent bilinear mask, which modifies Eq. (12) as

$$\mathbf{y} = (\mathbf{x} * \mathbf{k}'_{e1}) \odot \mathbf{W}_{e1} + (\mathbf{x} * \mathbf{k}'_{e2}) \odot \mathbf{W}_{e2} + \ldots + (\mathbf{x} * \mathbf{k}'_{ep}) \odot \mathbf{W}_{ep} \quad (13)$$

where $\mathbf{k}'_{ei}$ is the filter at ith branch and $\mathbf{W}_{ei}$, $1 \leq i \leq p$ are learnt non-negative SvId masks which sum to unity ($0 \preceq \mathbf{W}_{ei} \preceq 1$ and $\sum_{i=1}^{p} \mathbf{W}_{ei}(m,n) = \mathbf{1}$). Taking into consideration some desired properties for dynamic scene deblurring, we slightly modify the SvId-ASPP as follows. First, inverse filters for deblurring may require a very large receptive fields [21], [38], for which we cascade multiple stages (exactly three) of the given module. Second, as discussed previously, deblurring may also require unity receptive field for identity mapping, for which we consider the identity mapping as the first branch of each stage (instead of standard 3×3 filters in ASPP).

Remark 3: Our ASPP (Eq. (13)) admits SvId mapping with *diverse* receptive fields wherein each receptive field (other than 1 – the identity mapping) admits *numerous* filter realizations.

(Please refer to the Supplementary for a proof.) Intuitively, Eq. (13) through SvId masks $\mathbf{W}_{ei}$ realize numerous resultant filters which are linear combinations of constituent filters $\mathbf{k}'_{ei}$ of diverse receptive fields (see Fig. 6), independently in different spatial-coordinates (all of which are image-dependent).

The SvId masks are produced by a mask-generating network similar to the one employed for coherent-fusion, i.e., a cascade of 3×3 convolution layer, residual block, and another convolution layer (following [40]). The main difference is that we employ soft-max layer at the end (instead of sigmoid) as it has to produce multiple masks, each for individual branches, in contrast with a single mask in coherent fusion (please see our supplementary for more details). Figure 7 provides final SvId masks corresponding to individual branches, which reveals that the regions with large blurs (which typically happens at image edges or textured area) tend to have the large mask values for the branch with the large receptive fields (as desired [38]). This is in contrast with normal ASPP as used in DL-BMD method [40] where there exists *no* SvId masks; in other words, corresponding ASPP masks with respect to Figs. (b-e) in [40] are uniform irrespective of spatial regions and images (and hence does *not* possess SvId property).

V. Experiments

A. Implementation Details

To introduce our proposed techniques, we followed the generic DL-BMD network (Fig. 2, [40]), i.e., three stages each for encoder and decoder, and apart from our coherent fusion loss (Eq. (5)), we considered the MSE and perceptual losses. Our SvId ASPP consists of three stages, with sampling rates $\{1, 3, 5, 7\}$, $\{1, 4, 6, 8\}$, and $\{1, 2, 3, 4\}$. Since there exist no datasets for unconstrained DL dynamic scene blur, we contribute one with seven diverse exposure cases (of Fig. 5(a)), viz. $1:3$, $4:3$, $3:5$, $3:1$, $3:4$, $5:3$, & $1:1$. Following [15], exposure-overlap is randomly sampled from 10-100% with resolution-ratio 1:2. Our dataset is generated using the constrained DL blur dataset of DAVANET [40] and adapts it to the case of multi-exposure acquisition in dual-lens cameras (please refer to our Supplementary for the details on the dataset generation (Sec. S2c), data distribution (Table S2), network architecture (Fig. S2) and parameters (Sec. S2b)).

We optimize the computational cost as follows. We classify our training/testing dataset according to the optimal number of network levels each sample requires. For this, we find the registration error between the left- and right-view input images (following [25]). The ninety-quartile of the vertical displacement error (in pixels) is considered as the estimate of discrepancy, which is empirically selected for good classification accuracy. We do not consider horizontal displacement error because it can be primarily due to stereo parallax. Then an optimal decimation scale is chosen such that it reduces the maximum discrepancy of those samples within one pixel. For training, we confine all images in a batch to have a particular number of levels, thereby allotting optimal multi-scale network for each batch (which is derived from the single-scale network following Sec. III-B). For multi-scale case, weights in each scale are shared (Sec. III-B), and hence are updated together. For quantitative evaluation, we consider the mean absolute error (MAE) for disparity (lower values are better), and for deblurring, PSNR and SSIM in the view with relatively more degradation and respective offsets in the other view. A higher PSNR with small offsets implies good view-consistency.

B. Analysis and Discussions

1) Sensitivity to Resolution-Ratios and Noise: We analyse the performance of our network for different resolution-ratios, viz. 1:1 to 1:2.75 which span a wide range of present-day DL-cameras (in Fig. 8(b)). Note that the resolution-ratios and focal length ratios are synonymous [15], [32], and therefore findings from Fig. 8(b) hold good for focal length ratios as well. It is evident from the figure that though our network is trained for 1:2 case, the performance-decay is quite less over

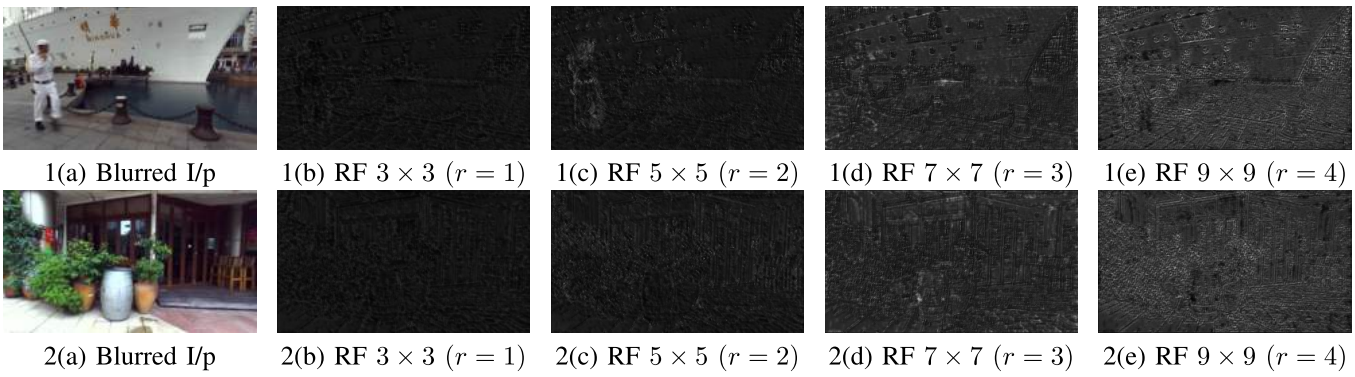
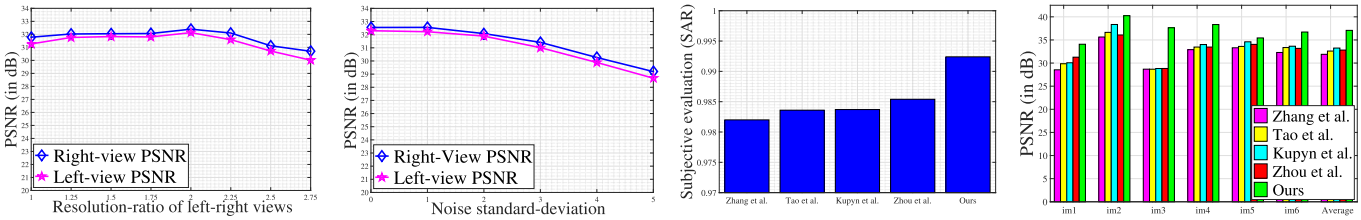


Fig. 7. Visualization of SvId-ASPP masks in different branches (RF is filter's receptive field and r is sampling rate). In Figs. (b-e), image regions with edges and textures which typically undergo higher degradations have larger receptive fields thereby allowing space-variant deblurring. Further, the masks vary with scene composition (compare Fig. 1 and Fig. 2), thereby admitting image-dependent deblurring.



(a) Resolution-ratio Analysis (b) Noise Analysis (c) SAR Analysis (d) SAR Implication

Fig. 8. Analysis: (a) Performance analysis for different resolution-ratios. (b) Analysis of image-noise (AWGN). (c) Evaluation of view-consistency using the subjective measure SAR [4]. (d) Application of view-consistency for DL super-resolution [31].

other cases, and further, view-consistency is well-preserved over the entire range (via close left-right PSNRs). This reveals the generalization capability of the coherent fusion module in channelling rich complementary features for diverse resolution and focal length ratios. Further, we analyse the effect of noise by introducing AWGN ($0 \leq \sigma \leq 5$ pixels) *independently* in the two views (in Fig. 8(a)). Note that over the entire σ range, the PSNRs are over 29 dB, and PSNR-differences are within 0.7 dB. This noise-robustness can be primarily attributed to the process of noise-addition during training (as in [40]). Note that our assumption of changing the resolution is equivalent to changing focal length is not strictly true for actual sensors with Bayer pattern and sensor noise. To account for this subtle inaccuracies, we consider real blurred images captured by actual unconstrained DL cameras (please see Sec. V-C).

2) Ablation Studies: To study the effectiveness of the proposed modules, we replace them with suitable analogous modules and retrain using the same strategy. Regarding coherent fusion, the major difference of our method from the only-existing DL-BMD network [40] is the view-aggregation at node-pair $\{A, B\}$ + self-supervision costs at $\{A, B\}$, and only self-supervision costs at node-pair $\{C, D\}$. Therefore to analyse the effect of coherent fusion, we remove those three components and considered view aggregation of [40] (at node-pair {C,D} in Fig. 2). In Table III, note the deviation by a large margin the deblurring performances in left- and right-views, which reveals significant view-inconsistency. This implies that information fusion seldom happens *without* coherent fusion modules, and here, the network tries

TABLE I

EFFECT OF SELF-SUPERVISION COSTS AT $\{C, D\}$ ON PSNR-OFFSETS AND NETWORK-SIZE FOR DIFFERENT UNCONSTRAINED DL EXPOSURE (EXP) RATIOS. (LOWER VALUES ARE BETTER.)

Approach	Exp 4:3	Exp 3:5	Exp 1:1	Size
W/o costs	1.1729 dB	1.1375 dB	0.3800 dB	5.98 M
W/ costs	0.8450 dB	1.0050 dB	0.2580 dB	5.98 M

to primarily improve the right-view PSNR (which is easy to accomplish due to its less information-loss), but neglects the left-view (where PSNR improvement is difficult due to more information-loss). Further, we analyse the effect of self-supervision costs at node-pair $\{C, D\}$ on view-consistency in Table I. Note that even though the costs at $\{C, D\}$ do not increase the number of training parameters, they provide a marginal improvement in view-consistency. To study the effect of adaptive scale-space, we considered only our fine-scale network for unconstrained DL configuration. Here, note that the MAE of disparities for different cases are quite high, which in turn leads to scene-inconsistent depth. We further study the effect of Eqs. (11) and (10) on multi-scale network, by retraining a similar network but with Eq. (10) using the same shared-weight strategy. It is clear from Table II that the multi-scale network with Eq. (11) performs better. Finally, we analyse our SvId ASPP for deblurring by replacing it with analogous ASPP module [3]. Table III reveals that the SvId ASPP significantly boosts the PSNRs as compared to the standard ASPP. Importantly, our method performs best in all aspects when all the three modules are present.

TABLE II
EFFECT OF EQS. (10) AND (11) IN OUR MULTI-SCALE NETWORK ON DISPARITY ERRORS AND NETWORK-SIZE. (LOWER VALUES ARE BETTER.)

Approach	Exp 4:3	Exp 3:5	Exp 1:1	Size
Eq. (10)	1.3186 px	2.0117 px	0.7615 px	5.98 M
Eq. (11)	0.8465 px	1.0043 px	0.7380 px	5.98 M

3) Importance of View-Consistency: To quantify the view-consistency of DL-BMD for stereoscopic applications, we consider the 'stereo-pairs accounting for rivalry' (SAR) metric [4] which measures the quality of DL images that have been afflicted by asymmetric distortions ($\leq$ 1 and higher values are better). Figure 8(c) compares SAR of state-of-the-art deep learning methods on the unconstrained dataset, which highlights the effectiveness of our proposal for those applications. For illustration, we employ those BMD methods as a preprocessing stage for a state-of-the-art DL super-resolution method [31], and quantify its performance in Fig. 8(d). Notably, there is a significant performance drop in competing methods (i.e., atleast by 4 dB), which is possibly because [31] is designed for view-consistent inputs akin to most DL-based works [2], [7], [11], but as revealed by SAR, the competing methods are ineffective in supplying to them view-consistent inputs.

4) View-Inconsistency via Bootstrapping: There exist DL applications that are *not* meant for stereoscopic vision, where it is desirable to have maximum deblurring performance in individual views. But due to training with the coherent fusion module, our method seldom raises the performance in one-view without respecting the other-view. So as to enable our network to cater to those kind of applications, we propose a bootstrapping approach where our *trained* network is fine-tuned *without* the self-supervision costs (L_{LR} and L_{RR}). Since we are starting with a well-balanced deblurring performance in the two-views, there exists a superior view-aggregation [40] provided by the already improved highly-degraded input image. This allows further restoration of blurred image that has relatively more image-features, while maintaining the superior performance of the other-view. Table III and Figs. 9–10 provide our bootstrapped results as well, which clearly reveals that our approach outperforms all other methods in this aspect too.

C. Comparisons

We consider for evaluation diverse DL-BMD methods, i.e., [40] that is designed for unconstrained DL and static scenes, and [15], [34] that handle constrained DL for dynamic scenes. We also include various state-of-the-art single-lens methods to represent scale-space approach [27], generative models [13], [22], and patch-based approach [37]. Following the comparison scheme of state-of-the-art DL-BMD [40], we fully fine-tune all deep learning based comparison methods on our new unconstrained DL dataset for fair comparison. Specifically, [13], [22], [27], [37], [40] were trained on our unconstrained DL dataset for 200K iterations with a batch size of 1. We also considered for evaluation constrained DL dataset of [40] and unconstrained DL static-scene dataset of [15]. Table III provides the quantitative evaluation of deblurring performance for different unconstrained DL settings. Our method has good PSNR in the left-right views with the least offset, which implies superior view-consistency. Note that even though the methods [13], [27], [37], [40] outperform ours for right-view images (i.e., PSNR+PS:Offset in Table III), those methods cannot perform well for the left-view mainly due to its relatively higher degradations (i.e., *both* motion blur and down-sampling), where our method showcase large gains. The reason is that, unlike our method (Sec. II-A), those methods lack a mechanism to transfer rich complimentary information from high-resolution to low-resolution/left-view images (which explains our higher left-view PSNR compared to other methods). Further, as discussed in Sec. V-B.4, the coherent fusion module equalizes the features of the two views for view-consistency, which limits our right-view PSNR to increase unconditionally without improving the other view (which explains our trade-off in right-view PSNR). Note that our bootstrap approach in Sec. V-B.4 addresses this trade-off (Table III). Also, it is evident from the Table that the competing methods produce a large discrepancy in disparities (e.g., MAE above two pixels in the exposure-case 3:5) whereas our scale-space approach is able to curb this effect. In terms of speed, our method is comparable to deep learning methods, and significantly better than model-based optimization methods [15], [34]. Averaging over all the seven cases, our method has a PSNR of 30.653 dB with an offset 0.706. More important, ours has a good margin over the next-best competitor [40] (27.718 dB with an offset 6.074). For qualitative evaluation, we provide two unconstrained examples with both left-right views (Fig. 9(1) and 10), and for all other examples we provide the left-view image with inset magnified left/right patches (with green/red boundary) from the highlighted image-region (in view of space). Nevertheless, we provide several unconstrained DL examples with both left-right views in our supplementary. It is evident from Fig. 9 that existing methods are *not* adequate for unconstrained DL dynamic scene deblurring, which calls for a new approach (like ours). We also evaluate our method on unconstrained DL static scene blur examples (from [15]) and constrained DL blur examples (from [40]) in Fig. 10 and demonstrate competitive performances. In summary, our method seamlessly addresses unconstrained DL dynamic scene deblurring under diverse exposures, exposure-overlap, and resolution-ratio; unconstrained static scene deblurring as in [15], and constrained DL deblurring as in [40].

We consider also for comparisons, real examples captured using actual unconstrained dual-lens cameras (Fig. 11(1-4)). Following [15], we employed the unconstrained DL set-up of Samsung S9+ smartphone to create this dataset, which provides a narrow- and wide-FOV pair with focal length 52 mm and 26 mm respectively, with former having twice the resolution. The examples include two well-lit scenarios (Fig. 11(1,4) and two-low-lit scenarios (Fig. 11(2,3). For comparison, we considered all the dual-lens deblurring methods ([15], [34], [40]) and two single-image deblurring methods ([22], [37]). Qualitative experiments in Fig. 11 reveal that our method produces good deblurring results overall, in particular,

TABLE III

QUANTITATIVE EVALUATIONS: SA - SCALE ADAPTIVE; CF - COHERENT FUSION; BS- BOOTSTRAP. (FIRST/SECOND)

Method	Xu et al. [34]	Mohan et al. [15]	Tao et al. [27]	Ramakrishnan et al. [22]	Kupyn et al. [13]	Zhang et al. [37]	Zhou et al. [40]	Ours	Ours (BS)	Ours W/o SA	Ours W/o CF	Ours W/o SvID
Unconstrained DL Case 2: Exposure 4:3												
MAE	1.929	2.273	1.9704	1.9195	1.9230	2.0488	1.9328	**0.8465**	0.8533	1.9718	0.8572	0.8318
PSNR	16.167	25.169	26.536	26.679	26.743	26.406	26.437	**30.581**	**30.560**	30.118	27.11	28.32
PS:Offset	0.8390	1.0600	6.4970	5.3718	5.7810	5.6060	5.6360	**0.8450**	**5.1810**	0.8971	5.2181	0.8677
SSIM	0.471	0.816	0.860	0.861	0.862	0.858	0.863	0.917	0.913	0.915	0.894	0.899
SS:Offset	0.0630	0.0130	0.0860	0.0644	0.0720	0.0730	0.0740	0.0070	0.0350	0.0081	0.0581	0.0083
Unconstrained DL Case 3: Exposure 3:5												
MAE	1.894	3.021	2.2524	2.2321	2.2204	2.3518	2.2444	**1.0043**	1.0066	2.2731	1.0076	1.0068
PSNR	17.465	26.348	26.593	26.425	26.536	26.431	26.040	**28.801**	**28.724**	28.402	26.181	27.65
PS:Offset	1.1230	1.9870	4.1550	4.0374	4.1280	3.2460	3.3640	**1.0050**	**4.1380**	1.1139	3.254	1.0178
SSIM	0.559	0.876	0.862	0.862	0.858	0.858	0.868	0.904	0.901	0.898	0.885	0.891
SS:Offset	0.0790	0.0140	0.0570	0.0260	0.0590	0.0470	0.0440	0.0090	0.0310	0.0131	0.0413	0.0454
Unconstrained DL Case 7: Exposure 1:1												
MAE	0.941	1.215	0.8869	0.8467	0.8794	0.9921	0.8672	**0.7380**	0.7718	0.7802	0.7591	0.7328
PSNR	17.047	26.815	26.984	26.024	26.906	25.854	29.198	**32.052**	**31.006**	31.047	29.187	28.180
PS:Offset	1.4260	1.8090	1.2520	1.2016	1.3060	1.2960	3.9320	**0.2580**	**2.3430**	0.2577	3.897	0.357
SSIM	0.477	0.854	0.861	0.824	0.855	0.828	0.892	0.905	0.904	0.905	0.889	0.857
SS:Offset	0.0940	0.0300	0.0330	0.0284	0.0310	0.0330	0.0480	0.0070	0.0350	0.0065	0.0493	0.0073
Time (S)	2160	1630	0.507	0.37	0.39	0.524	0.31	0.34/scale	0.34/scale	0.34/scale	0.33/scale	0.31/scale
Size (M)	-	-	8.06	4.17	5.09	21.69	4.83	5.98	5.98	5.98	5.90	5.93

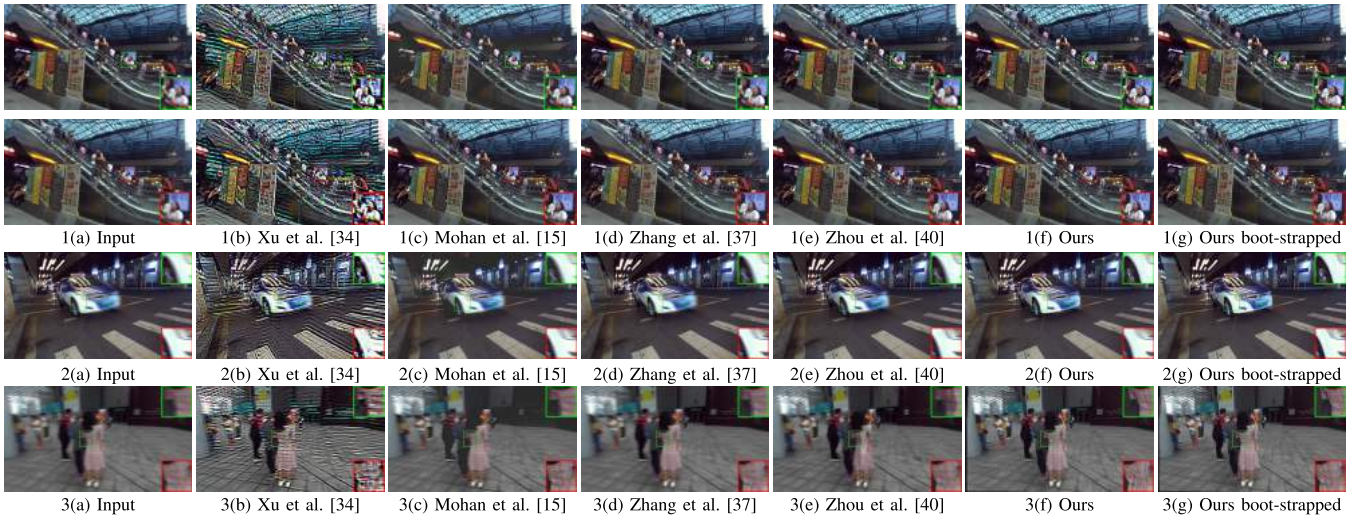


Fig. 9. Comparisons for unconstrained DL configuration (examples 1 - 3:5 case, 2 - 4:3 case, and 3 - 1:1 case. (First example provides left-view (first-row) and right-view individually, whereas remaining examples provides left- and right-view patches inset. Our method is able to produce view-consistent results as compared to the competing methods (compare patches from *both* views). After bootstrapping (1(g) and 3(g)), our method produces good view-*in*consistent results as well (please use high-resolution displays for best view).

Fig. 10. Comparisons for constrained DL dynamic scene blur case (from [40]) and unconstrained DL static scene case (from [15]). Our method produces the best view-consistent results, and our deblurring performance is comparable with the state-of-the-art methods.

quite good restoration for the left-view which is degraded by both low-resolution as well as motion blur, and produces good view-consistencies between the left-right views as compared to the competing methods.

(Please refer to our supplementary for training details and cost analysis, details of dataset generation, extensive comparisons of Figs. 9–10, remaining results of Table III (Cases 1, 4-6), SSIM evaluations for Sec. V-B.1, and

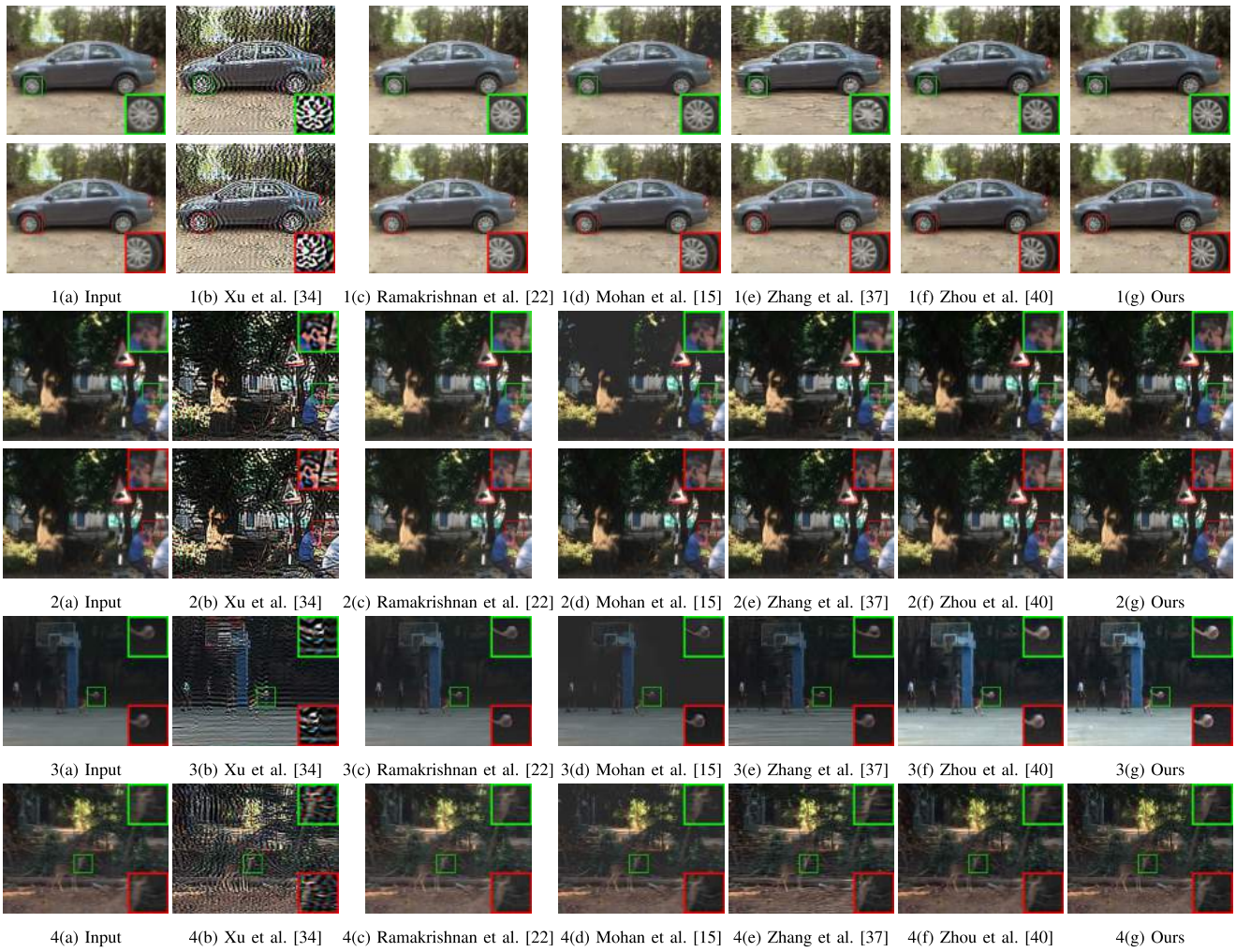


Fig. 11. Qualitative comparisons with real unconstrained DL images: Examples 1 and 4 depict well-lit scenarios, whereas examples 2 and 3 depict low-lit scenarios. First two examples provide left-view (first-row) and right-view individually, whereas remaining examples provide left- and right-view patches inset. Note that our method exhibits quite good deblurring performance, especially for the left-view which is degraded by both low-resolution as well as motion blur, and produces good view-consistencies as compared to the competing methods.

qualitative results of the SAR implication in Sec. V-B.3. Further, we provide additional evaluations in DL-BMD datasets of ours, [40], and [15].)

Limitations: Extensive quantitative and qualitative evaluations of state-of-the-art BMD methods on our unconstrained DL blur dataset reveal that the low-resolution/left-view images (which incur degradation due to *both* motion blur and downsampling) admit significantly less information-retrieval performance as compared to the high-resolution images (which have *only* motion blur). This motivated us to employ the high-resolution image as the reference to transfer complimentary information (Sec. II-A). However, there exist scenarios where this scheme is not effective. For example, consider an over-lit (or low-lit) scene wherein the high-resolution image is highly over-exposed (or underexposed) as compared to the low-resolution image. Note that our dataset does not contain such extreme cases, as in standard motion blur datasets [16], [40], and our method is not applicable to those scenarios.

VI. Conclusion and Future Works

In this paper, we proposed the first dynamic scene deblurring method for present-day unconstrained DL cameras and addressed its three major problems: First, we dealt with the inherent view-consistency problem using a Coherent Fusion Module. Next we addressed the problem of scene-inconsistent disparity via an Adaptive scale-space approach. This enables effectively adapting a standard deep learning network for different image-scales. Finally, we tackled the space-variant image-dependent nature of dynamic scene blur by proposing an advanced ASPP module. We also contributed a new dataset for the current problem. Various evaluations with the state-of-the-art methods reveal the importance of our method.

Several deep learning based AR/VR applications using DL cameras, e.g., super-resolution [11], [31] and style-transfer [2], [7], directly applied to unconstrained DL set-up naturally leads to view-*in*consistency. Our Coherent Fusion Module can potentially extend these methods to tackle this problem. Second, our Adaptive Scale-space Approach can potentially allow existing deep-learning networks meant for diverse applications to effectively accommodate low-resolution images (e.g., [40] in Fig. 5(c)). Further, the ASPP [3] is widely-used in semantic segmentation, object detection, and visual question answering. But SvId nature is inherent in these applications as well, e.g., spatial-locations and scales of semantic objects

in an image can freely vary, and these attributes are image-dependent. Therefore, another exciting research direction is to explore the potential of SvId-ASPP for those applications.

REFERENCES

[1] M. Bätz, T. Richter, J.-U. Garbas, A. Papst, J. Seiler, and A. Kaup, "High dynamic range video reconstruction from a stereo camera setup," *Signal Process., Image Commun.*, vol. 29, no. 2, pp. 191–202, Feb. 2014.

[2] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6654–6663.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[4] M.-J. Chen, C.-C. Su, D.-K. Kwon, L. K. Cormack, and A. C. Bovik, "Full-reference quality assessment of stereopairs accounting for rivalry," *Signal Process., Image Commun.*, vol. 28, no. 9, pp. 1143–1155, Oct. 2013.

[5] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Proc. 8th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Jun. 2016, pp. 1–6.

[6] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3848–3856.

[7] X. Gong, H. Huang, L. Ma, F. Shen, W. Liu, and T. Zhang, "Neural stereoscopic image style transfer," in *Proc. Eur. Conf. Comput. Vis.* Springer, Sep. 2018, pp. 54–69.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry In Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[9] M. Innmann *et al.*, "NRMVS: Non-rigid multi-view stereo," 2019, *arXiv:1901.03910*. [Online]. Available: http://arxiv.org/abs/1901.03910

[10] G. Iyer, J. K. Murthy, G. Gupta, K. M. Krishna, and L. Paull, "Geometric consistency for self-supervised end-to-end visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 267–275.

[11] D. S. Jeon, S.-H. Baek, I. Choi, and M. H. Kim, "Enhancing the spatial resolution of stereo images using a parallax prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1721–1730.

[12] M. Jin, G. Meishvili, and P. Favaro, "Learning to extract a video sequence from a single motion-blurred image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6334–6342.

[13] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192.

[14] J. Mo and J. Sattar, "DSVO: Direct stereo visual odometry," 2018, *arXiv:1810.03963*. [Online]. Available: http://arxiv.org/abs/1810.03963

[15] M. M. M. R, S. Girish, and R. Ambasamudram, "Unconstrained motion deblurring for dual-lens cameras," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7870–7879.

[16] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.

[17] A. Oppenheim and R. Schafer, *Discrete Time Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2014.

[18] L. Pan, Y. Dai, M. Liu, and F. Porikli, "Simultaneous stereo video deblurring and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6987–6996.

[19] W.-J. Park, S.-W. Ji, S.-J. Kang, S.-W. Jung, and S.-J. Ko, "Stereo vision-based high dynamic range imaging using differently-exposed image pair," *Sensors*, vol. 17, no. 7, p. 1473, Jun. 2017.

[20] N. F. Pashchenko, K. S. Zipa, and A. V. Ignatenko, "An algorithm for the visualization of stereo images simultaneously captured with different exposures," *Program. Comput. Softw.*, vol. 43, no. 4, pp. 250–257, Jul. 2017.

[21] K. Purohit and A. N. Rajagopalan, "Region-adaptive dense network for efficient motion deblurring," in *Proc. AAAI*, 2020, pp. 1–8.

[22] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman, "Deep generative filter for motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2993–3000.

[23] A. Sellent, C. Rother, and S. Roth, "Stereo video deblurring," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 558–575.

[24] X. Shen, H. Gao, X. Tao, C. Zhou, and J. Jia, "High-quality correspondence and segmentation estimation for dual-lens smart-phone portraits," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3277–3286.

[25] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1279–1288.

[26] N. Sun, H. Mansour, and R. Ward, "HDR image construction from multi-exposed stereo LDR images," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2973–2976.

[27] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.

[28] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich, "Examining the impact of blur on recognition by convolutional networks," 2016, *arXiv:1611.05760*. [Online]. Available: http://arxiv.org/abs/1611.05760

[29] J. Wang, T. Xue, J. T. Barron, and J. Chen, "Stereoscopic dark flash for low-light photography," 2019, *arXiv:1901.01370*. [Online]. Available: http://arxiv.org/abs/1901.01370

[30] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[31] L. Wang *et al.*, "Learning parallax attention for stereo image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12250–12259.

[32] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 168–186, Jun. 2012.

[33] R. Xiao, W. Sun, J. Pang, Q. Yan, and J. Ren, "DSR: Direct self-rectification for uncalibrated dual-lens cameras," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 561–569.

[34] L. Xu and J. Jia, "Depth-aware motion deblurring," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, Apr. 2012, pp. 1–8.

[35] B. Yan, C. Ma, B. Bare, W. Tan, and S. Hoi, "Disparity-aware domain adaptation in stereo image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13179–13187.

[36] F. Zhang and F. Liu, "Casual stereoscopic panorama stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2002–2010.

[37] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5978–5986.

[38] J. Zhang *et al.*, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2521–2529.

[39] K. Zhang *et al.*, "Deblurring by realistic blurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2737–2746.

[40] S. Zhou, J. Zhang, W. Zuo, H. Xie, J. Pan, and J. S. Ren, "DAVANet: Stereo deblurring with view aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10996–11005.

Deep Dynamic Scene Deblurring for Unconstrained Dual-lens Cameras (Supplementary Material)

Mahesh Mohan M. R., Nithin G. K., A. N. Rajagopalan, *Senior Member, IEEE*

The contents of this supplementary material are arranged as follows:

1) Justifications
   a) Justification that the coherent fusion module imposes Condition A and B.
   b) Justification of Remark 2 (i.e., as compared to using the same network as a black-box to process lower image-scales as in Eq. (10), the network mapping in Eq. 11 is optimal.)
   c) Justification of Remark 3 (i.e., our ASPP in Eq. (13) admits SvId mapping with *diverse* receptive fields and admits numerous filter realizations).
2) Analysis and Discussions
   a) Inadequacy of the DL Prior in [5] for dynamic scene DL-BMD (from Sec. III-A).
   b) Implementation details and Cost analysis (extension of Sec. V-A).
   c) Dataset generation.
   d) Extensive evaluation (i.e., Cases 1, 4-6 of Table III, SSIMs of Sec. V-B1, and detailed comparisons of Figs. 9–10)
   e) Qualitative comparisons of DL-BMD preprocessing for DL super-resolution [16] (Sec. V-B3).
   f) Additional evaluations in our unconstrained-DL dynamic scene blur dataset (with detailed comparisons).
   g) Visualization of left-view and right-view images for our unconstrained-DL dynamic scene blur dataset.
   h) Additional evaluations in unconstrained-DL static scene blur dataset [5] (with detailed comparisons).
   i) Additional evaluations in constrained-DL dynamic scene blur dataset [22] (with detailed comparisons).

Note: The sections, equations and figures in the supplementary are indexed with a prefix 'S' (e.g., Eq. (S3), Sec. S4).

S1. Justifications of Remarks 1-3

(a) **Remark 1:** Given that the costs $\{L_{LR}, L_{RR}\}$ are imposed, the mapping of Eqs. (2)-(4) *minimizes* individual costs L_{LR} and L_{RR}. Further, both costs L_{LR} and L_{RR} are necessary to satisfy the conditions: (A) The nature of output signals in the left-right views are *identical*; (B) *Both* the outputs exhibit the properties of the input with *higher* information.

Justification: (For the sake of simplicity, we first assume that occlusions and specularities in DL images are negligible.) To justify the first part, it is sufficient to show that there exists atleast a case where the mapping in Eqs. (2)-(4) attains the least objective (zero) for the costs L_{LR} and L_{RR} (as $L_{LR}, L_{RR} \geq 0$). It is evident from Eqs. (2)-(4) that the cost $L_{LR} = 0$ when $\mathbf{W}^L = \mathbf{W}^R = \mathbf{1}$, and $L_{RR} = 0$ when $\mathbf{W} = \mathbf{W}^R = \mathbf{1}$. To justify the second part, we show that criteria that minimize L_{LR} satisfies the Condition A but *not* necessarily Condition B; similarly, criteria of L_{RR} satisfies the Condition B but *not* necessarily Condition A, and finally, the common criterion of L_{LR} and L_{RR} satisfy both Conditions A and B. For brevity, we refer to "first-view is sourced by second-view" if the first-view's output is formed by the second-view's input. From Eq. (5), the cost $L_{LR} = 0$ implies $\mathbf{y}^R = \mathbf{y}^L$, (i.e., Condition A). For non-identical left-right inputs in general, the cost $L_{LR} = 0$ when the left- and right-views are sourced by only right-view (i.e., $\mathbf{W} = \mathbf{W}^L = \mathbf{1}$ in Eqs. (2)-(4)), or only left-view (i.e., $\mathbf{W} = \mathbf{0}, \mathbf{W}^R = \mathbf{1}$), or a combination of left- and right-view (i.e., $\mathbf{0} \prec \mathbf{W} \prec \mathbf{1}, \mathbf{W}^L = \mathbf{W}^R = \mathbf{1}$). Clearly from Eq. (5), $L_{RR} > 0$ for the last two cases, as $\mathbf{y}^R \neq \mathbf{x}^R$ (thereby violating Condition B). Similarly, $L_{RR} = 0$ implies $\mathbf{y}^R = \mathbf{x}^R$, (i.e., condition B). The cost L_{RR} can be zero when right-view, but not left-view, is sourced by right-view (i.e., $\mathbf{W} = \mathbf{W}^R = \mathbf{0}$) or both right and left view are sourced by the right view (i.e., $\mathbf{W} = \mathbf{W}^L = \mathbf{1}$). For the first case $L_{LR} > 0$, as $\mathbf{y}^R \neq \mathbf{y}^L$ (thereby violating Condition A). Resultantly, the common criterion that minimizes L_{LR} and L_{RR} is when *both* the left- and right-views are sourced by only right-view (which satisfies both Conditions A and B), which proves the remark. (Relaxing the assumption on occlusions and specularities, the word "sourced" becomes "predominantly sourced", wherein occlusion and specularity information will be passed to left- and right-view outputs by the left-and right-view inputs, respectively.) Also, note that without the coherent fusion module, the masks $W_L = W_R = 0$ (i.e., the center-view ceases to transfer complimentary information). It is evident from the above discussion that in this case $L_{LR} > 0$ and $L_{RR} > 0$, and hence do not allow the minima ($= 0$).

Mohan M. R., Nithin G. K. and A. N. Rajagopalan are with the Department of Electrical Engineering, Indian Institute of Technology Madras, India. e-mail: {ee14d023, ee15b047, raju}@ee.iitm.ac.in.

(b) **Remark:** As compared to using the same network as a black-box to process lower image-scales as in Eq. (10), the network mapping in Eq. 11 is optimal.

We briefly review some signal processing concepts used here [8]. For sake of simplicity, we consider one-dimensional signal representation. We denote the frequency spectrum of a discrete signal $\mathbf{x}(n)$ by $\mathbf{X}(\omega)$ (where ω is the frequency domain). The convolution of $\mathbf{x}(n)$ and $\mathbf{y}(n)$ (denoted by $\mathbf{x}(n) * \mathbf{y}(n)$) results in a frequency spectrum $\mathbf{X}(\omega)\mathbf{Y}(\omega)$. Decimating a signal $\mathbf{x}(n)$ by a factor D first removes its high-frequency content (via anti-aliasing filter) and then *expands* the frequency spectrum by D. In contrast, interpolating a signal $\mathbf{x}(n)$ first *compresses* the frequency spectrum by D and then removes its high-frequency content (via anti-imaging filter). We denote the decimation and interpolation by $\mathbf{X}_{\downarrow D}$ and $\mathbf{X}_{\uparrow D}$, respectively. In general, decimation followed by an interpolation (i.e., $(\mathbf{X}_{\downarrow D})_{\uparrow D}$) is *not* an inverse operation due to anti-aliasing operation. For brevity, we refer to "a particular feature of $\mathbf{X}(\omega)$ matches $\mathbf{Y}(\omega)$" if that feature of $\mathbf{X}$ is present in $\mathbf{Y}$ as it is or as in a decimated form.

We next provide an outline of our proof. In a scale-space network, the output of lower-scale network should be the decimated version of the fine-scale network-output [2], [15], [6]. This implies that frequency spectrum of lower-scale network-output must *match* to that of fine-scale network-output (except at those high-frequency features lost due to decimation). Our DL deblurring network maps the input via a composition of individual functions, which are realized using a cascade of convolutions, non-linearities (Leaky ReLu), decimations in encoder and interpolations in decoder, etc. Resultantly, for output features of lower-scale networks to match that of fine-scale network, all those individual functions must map to *matching* features for *all* image-scales. However, we show (in Remark 2) that directly employing the fine-scale network in lower-scales maps to *complimentary* features for convolutions, and hence fails to produce matching features in lower-scales. Further, we show that our proposed transformation alleviates the aforementioned issue of convolutions. Finally, we show that this property of transformation generalizes to other network-functions as well, as required in a scale-space network.

We first consider the case of convolution, which maps an input tensor $\mathbf{f}$ with depth d to a tensor $\mathbf{g}$ with depth d' as

$$\mathbf{g}^j = \sum_{i=1}^{d} \mathbf{f}^i * \mathbf{h}^{\{i,j\}} \quad : \ 1 \leq j \leq d', \tag{S1}$$

where $\mathbf{g}^j$ (the jth layer of $\mathbf{g}$) is produced by aggregating the convolution of $\mathbf{f}^i$ and filters $\mathbf{h}^{\{i,j\}}\ \forall i$. In particular, $\mathbf{f}$ in a decoder stage is obtained by concatenating feed-forward encoder features and decoder features [22], whereas the standard residual block (i.e., $\mathbf{g}^j = \sum_{i=1}^{d} \mathbf{f}^i * \mathbf{h}^{\{i,j\}} + \mathbf{f}^i$) and atrous spatial pyramid pooling (i.e., $\mathbf{g}^j = \sum_{i=1}^{d}\sum_{k=1}^{p} \mathbf{f}^i * \mathbf{h}^{\{i,j,k\}}$) [1] has equivalent convolution filter as $\overline{\mathbf{h}}^{\{i,j\}} = \mathbf{h}^{\{i,j\}} + \delta(i,j)$ and $\overline{\mathbf{h}}^{\{i,j\}} = \sum_{k=1}^{p} \mathbf{h}^{\{i,j,k\}}$, respectively.

Remark 2: Convolution filters optimized to map certain frequency features in the fine-scale (Eq. (9)) need *not* map to matching features in lower scales (Eq. (10)), whereas with the transformation in Eq. (11) map to matching features.

Justification: The mappings in both Eqs. (10) and (11) employ identical convolution filter in fine-scale as well as lower scales (via the same $T(\cdot)$), but the difference is that for lower-scales the former gets a decimated input, whereas the latter gets an interpolated version of the decimated input (i.e., with the same resolution of fine-scale). In frequency domain, convolution mapping (Eq. (S1)) in the fine-scale is

$$\mathbf{G}^j(\omega) = \sum_{i=1}^{d} \mathbf{F}_i(\omega)\mathbf{H}^{\{i,j\}}(\omega), \tag{S2}$$

where $\mathbf{H}_{\{i,j\}}(\omega)$ is the frequency spectrum of convolution filter. The spectrum $\mathbf{H}_{\{i,j\}}(\omega)$ is typically non-uniform [19] and hence maps/extracts frequency features in a specific way (e.g., in Fig. 5(1), the filter extracts desired features while suppressing undesired features). Now we consider the case of using the same operation of Eq. (S2) for lower scales (i.e., Eq. (10)). This results in

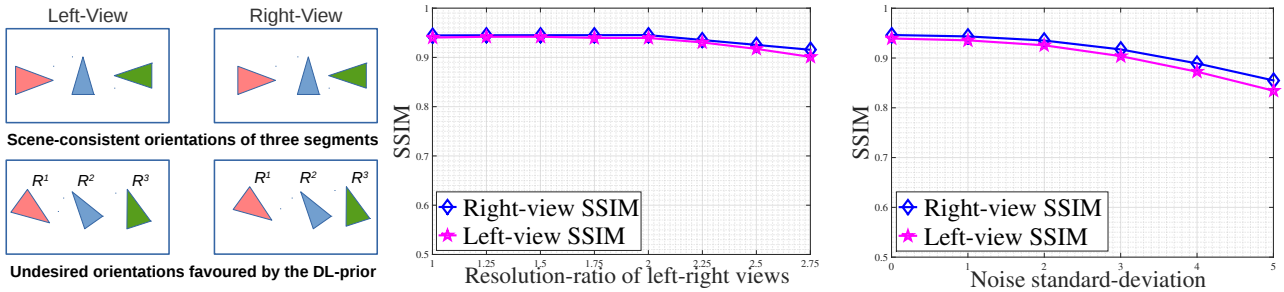
$$\mathbf{G'_D}^j(\omega) = \sum_{i=1}^{d} \mathbf{F}_i(\omega)_{\downarrow D}.\mathbf{H}^{\{i,j\}}(\omega), \tag{S3}$$

Note that the spectrum $\mathbf{H}_{\{i,j\}}(\omega)$ now maps/extracts frequency features in a different way as compared to Eq. (S2) (due to expanded input spectrum). Resultantly, it can map to *non*-matching features in lower scales (e.g., see Fig. 5(2) where the desired features get suppressed). Next, we consider the proposed transformation (Eq. (11)):

$$\mathbf{G_D}^j(\omega) = \sum_{i=1}^{d} \mathbf{F}_i(\omega)_{\downarrow D\uparrow D}.\mathbf{H}^{\{i,j\}}(\omega). \tag{S4}$$

Note that we do not consider the overall decimation of Eq. (11) as it is *not* present after each convolution stage (but *only once* at the network-output). In Eq. (S4), frequency features of input coincide with that of the fine-scale input in frequency domain (due to inverse-scaling or $\uparrow D$). Resultantly, the spectrum $\mathbf{H}_{\{i,j\}}(\omega)$ maps/extracts frequency features in the same way as compared to Eq. (S2), and hence map to matching features (except at those high-frequency features lost due to anti-aliasing) as warranted by scale-space network. This is illustrated in Fig. 5 (compare Figs. (2) and (3)).

Remark 2 establishes that our transformation in lower-scales maps to matching features for convolution stages. We next show that this transformation is favourable in other network-stages as well, which ensures its applicability for lower-scales.



(a) DL-prior [5] Analysis (b) Resolution-ratio Analysis (SSIM) (c) Noise Analysis (SSIM)

Fig. S1: (a) Due to possibly different relative-motions in individual dynamic objects, the pose-ambiguity of DL-prior [5] need *not* be identical in different objects. The figure shows the case of different in-plane rotation ambiguity ($\{\mathbf{R}^1, \mathbf{R}^2, \mathbf{R}^3\}$) in three different objects, which clearly derails the scene-consistency as required for most DL applications. (b) Performance analysis for resolution-ratios (SSIM). (c) Performance analysis of image-noise (SSIM).

Generalization of Remark 2: The non-linearities, such as bias, leaky ReLU, etc., optimized for fine-scale is applicable to lower scales under the input-feature transformation of Eq. (11).
Justification: The non-linear transformations such as bias, leaky ReLU, etc., are point-wise operations and these functions are continuous (in particular, bias is a constant point-wise offset and leaky ReLU is continuous though *not* differentiable at origin). Further, as input images have predominantly low-frequency components, which is supported by the natural image priors such as total variation [11], l_0 in image-gradients [20], dark-channel [9], etc., input features to these non-linearities are predominantly low-pass. Therefore, the point-wise values of decimated (by D) and then interpolated (by D) version will have closer intensity values as that of the fine-scale image. Hence, due to closer point-wise values and continuous characteristic of non-linear functions, the response of the non-linearities to decimated-and-then-interpolated input must be closer to the corresponding response of the fine-scale input. Finally, under the assumption that anti-aliasing and anti-imaging filters are ideal, intermediate stages of decimation (in encoder) and interpolation (in decoder) map to matching features as that of fine-scale network. Therefore, with the proposed transformation of Eq. (11), fine-scale network in lower scales map to matching features as that of fine-scale (as required in a scale-space network).

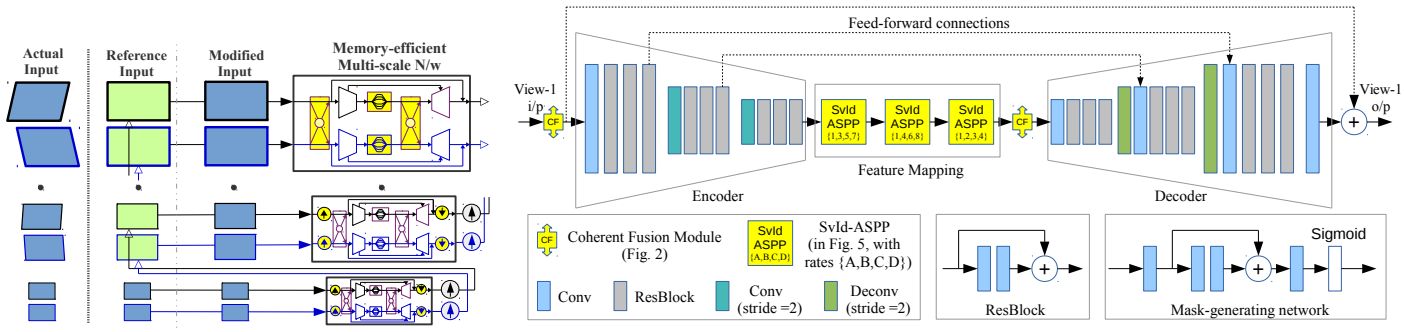
(c) **Remark 3:** Our ASPP (Eq. (13)) admits SvId mapping with *diverse* receptive fields wherein each receptive field (other than 1 – the identity mapping) admits *numerous* filter realizations.
Justification: The resultant filter in Eq. (13) at spatial location $\{m, n\}$ can be represented as

$$\mathbf{y} = \mathbf{x} * \mathbf{k}'(m,n), \quad \text{where } \mathbf{k}'(m,n) = \mathbf{W}_{e1}(m,n) \cdot \mathbf{k'}_{e1} + \mathbf{W}_{e2}(m,n) \cdot \mathbf{k'}_{e2} + \cdots + \mathbf{W}_{ep}(m,n) \cdot \mathbf{k'}_{ep} \tag{S5}$$

As the masks $\mathbf{W}_{ei}(m,n),\ 1 \leq i \leq p$ (in Eq. (13)), that linearly combine filter basis $\mathbf{k'}_{ei}$, can vary with spatial-coordinates (m,n), our modified ASPP can produce different filters at different spatial locations. Further, as those masks are function of input blurred image (via mask-generating network), the modified ASPP becomes image-dependent as well, and hence admits SvId mapping. Denoting the receptive field of a filter $\mathbf{k}'$ as $R(\mathbf{k}')$, the receptive field for the modified ASPP at a coordinate (m,n) becomes $\max R(\mathbf{k'}_{ei})$ if $\forall j > i, \mathbf{W}_{ej}(m,n) = 0$, and hence the image-dependent masks diversifies the receptive field (Fig. 6(b)-bottom). Further, a resultant filter with receptive field $R(\mathbf{k'}_{ei})$ can be realized by numerous linear combination of filters $\mathbf{k'}_{ej} : j \leq i$, specifically, in Eq. (S5) filters obtained by all possible combinations for $\mathbf{W}_{ek}(m,n) : k < i$ with $\mathbf{W}_{ei}(m,n) > 0$ and $\mathbf{W}_{ek}(m,n) = 0 : k > i$. Also, note that even though the individual filter-basis ($\mathbf{k}_{ei}$) is sparse, this linear combination of multiple filter-basis can produce dense filter for a given receptive field. Finally, a cascade of M such modules for deblurring retains the SvID property, and further increases the receptive field to $\{\sum_{i=1}^{M} R(\mathbf{k'}^{(i)})\} - M + 1$, where $\mathbf{k'}^{(i)}$ is the effective filter (Eq. (S5)) at the ith stage (because $R(\mathbf{k'_1} * \mathbf{k'_2}) = R(\mathbf{k'_1}) + R(\mathbf{k'_2}) - 1$ [8]).

S2. Analysis and Discussions

(a) **Inadequacy of the DL-prior [5]**: Dual-lens prior in [5] is *not* directly applicable in producing scene-consistent disparities in dynamic-scene deblurring for unconstrained DL.
Justification: First, we summarize the working principle of DL-prior. According to Claim 1 in [5], there exist multiple valid solutions of MDF-pairs (that quantify camera-motion) for image-pair in the left-right views, wherein some solutions produce *scene-inconsistent* disparities. Assuming that motion blur is due to *only* camera motion, the DL prior in [5] promotes a valid MDF-pair, but which can only resolve the deblurred image upto an unknown pose-variation of the actual scene (which is denoted by $\mathbf{R}_n$). To aid a fair evaluation, we allow the following relaxations in [5]: (a) Despite [5] restricts to only 3D rotation-changes due to only camera-motion assumption, we assume that [5] handles 3D translations as well, which is required to model dynamic scene blur [10], [13]. (b) Even though extending [5] for dynamic scene deblurring is non-trivial, as it involves



(a) Our Memory-efficient Multi-scale Network (b) Our Fine-scale Network

Fig. S2: Network Architecture: (a) Our multi-scale network is obtained by reusing the same network (our fine-scale network) using the proposed transformation (highlighted in yellow). (b) Our fine-scale network consists of a three-stage encoder/decoder, with SvId employed for feature mapping and coherent fusion module to balance signals in the two-views. The same network is shared for both views.

complex pipeline which include coherently segmenting dynamic objects in the two views and stitching different segments with negligible artifacts in seams [10], [13], [18], we assume that such a pipeline exists. Now we proceed to the justification. The ambiguity due to $\mathbf{R}_n$ causes a relative change in scene-orientation in both the views. Though it does *not* produce any issues for the case of static scene (as it renders the entire scene to have an arbitrary pose-change), this is not the case for dynamic scenes where each dynamic object can have (relative) independent motion. Let there be x dynamic objects, then the DL-prior on individual segments produces n independent pose-ambiguity (say $\mathbf{R}_n^i,\ 1 \leq i \leq x$) as the MDFs in each segments can be unrelated. Resultantly, it renders individual objects in the scene to have different pose-changes, e.g., as illustrated in Fig. S1(a), a horizontally moving object, with respect to background, can be rendered moving diagonally due to an in-plane rotational ambiguity (as considered in Fig. (2) of [5]), which clearly distorts scene-consistent disparities.

(b) **Implementation details and Cost analysis**: Figure S2 provides our multi-scale and fine-scale deblurring networks. Note that the proposed techniques are highlighted in yellow, and both views share the same architecture. As shown in Fig. S2(a), our multi-scale network is generated by reusing the fine-scale network, but with the transformation in Eq. (11). Our fine-scale network (Fig. S2(b)) consists of three encoder-decoder stages with SvId-ASPP for feature-mapping, and coherent fusion module to balance the nature of left-right view inputs (for view-consistency). To control possible intensity variations between the left- and right-view features, we normalize the mean and standard deviation of the left-view input of the coherent fusion module to that of its right-view input. We employ leaky-ReLU as the non-linearity, and all filter-weights are selected as 3×3 for memory considerations. For disparity estimation, we employ an off-the-shelf disparity estimation network of [22]. We do note that since our method requires *only* disparity-maps (unlike [22] which warrants network-specific disparity-features too), any disparity estimation methods irrespective of conventional and deep learning based can be employed in our method. To create bilinear masks, we consider the light-weight network shown in Fig. S2(b) with a soft-max layer at the end to normalize multiple masks (of SvId-ASPP). The input to the mask-generating network of coherent fusion module is the difference of the registered input two-view features. We employ independent mask-generating networks for coherent fusion module and SvId-ASPP due to their different functions.

For training our deblurring network, apart from the self-supervision costs of coherent fusion module (Eq. (5)), we consider two supervision costs to measure the difference between the deblurred images ($\{\hat{\mathbf{L}}^L, \hat{\mathbf{L}}^R\}$) and sharp images ($\{\mathbf{L}^L, \mathbf{L}^R\}$) for the left-right views (as discussed in Sec. III-B, we average these losses over all network-levels.) The first one is an objective cost based on the standard MSE loss:

$$L_{mse} = \frac{1}{S}\sum_{l=1}^{S}\frac{1}{2CMN}\sum_{k\in\{L,R\}}\|\hat{\mathbf{L}}_l^k - \mathbf{L}_l^k\|_2, \tag{S6}$$

where S is the number of scale-space network-levels, $\mathbf{L}_l$ denotes image at level l, and C, M, and N are dimensions of image. The second cost is the perceptual loss employed in [22], which is the l_2-norm between conv3-3 layer VGG-19 [14] features of deblurred images and sharp images:

$$L_{vgg} = \frac{1}{S}\sum_{l=1}^{S}\frac{1}{2C_vM_vN_v}\sum_{k\in\{L,R\}}\|\Phi_{vgg}(\hat{\mathbf{L}}_l^k) - \Phi_{vgg}(\mathbf{L}_l^k)\|_2, \tag{S7}$$

where Φ_{vgg} is the required VGG mapping, and C_v, M_v, and N_v are dimensions of VGG features. Denoting the normalized coherent fusion cost in Eq. (5) as L_{cf}, our overall loss function is empirically selected as $0.4L_{cf} + 0.5L_{mse} + 0.1L_{vgg}$. Our

TABLE S1: Quantitative evaluations: SA - Scale adaptive; CF - Coherent fusion; BS- Bootstrap. (First/Second)

Method	Xu et al. [18]	Mohan et al. [5]	Tao et al. [15]	Ramakrishnan et al. [12]	Kupyn et al. [3]	Zhang et al. [21]	Zhou et al. [22]	Ours	Ours (BS)	Ours W/o SA	Ours W/o CF	Ours W/o SvID
Unconstrained DL Case 1: Exposure 1:3												
MAE	1.9071	1.3504	1.8256	1.7762	1.7738	1.9312	1.7658	**0.7718**	0.7838	1.7782	0.7846	0.7952
PSNR	22.118	27.724	27.844	27.952	27.938	27.665	28.102	**30.132**	**30.127**	29.852	28.456	29.177
PS:Offset	1.7510	2.6440	6.3310	7.4838	7.2230	5.4890	6.0460	**0.1580**	**6.3260**	0.2532	6.1211	0.2541
SSIM	0.53	0.888	0.874	0.883	0.873	0.871	0.890	0.915	0.895	0.908	0.900	0.899
SS:Offset	0.1280	0.0070	0.0820	0.0789	0.0870	0.0770	0.0680	0.0030	0.0290	0.0080	0.0710	0.0076
Unconstrained DL Case 4: Exposure 3:1												
MAE	2.1500	2.4130	1.7118	1.6926	1.6711	1.7940	1.6991	**0.7277**	0.7301	1.7754	0.7431	0.7517
PSNR	17.958	28.79	27.246	27.432	27.360	26.007	26.991	**31.762**	**31.027**	29.971	27.357	28.177
PS:Offset	1.446	3.261	11.20	12.568	17.61	12.430	20.832	**0.6580**	**18.663**	0.7574	17.683	0.8531
SSIM	0.447	0.831	0.863	0.872	0.863	0.862	0.857	0.963	0.957	0.9228	0.877	0.902
SS:Offset	0.0932	0.0505	0.1252	0.1148	0.1324	0.1192	0.1309	0.0120	0.0652	0.0216	0.0581	0.0238
Unconstrained DL Case 5: Exposure 3:4												
MAE	1.965	1.2340	1.9217	1.8620	1.8645	2.0192	1.8837	**0.8716**	0.9142	1.9218	0.8872	0.8712
PSNR	14.317	27.725	27.303	27.511	27.415	27.191	27.426	**30.989**	**30.402**	30.343	27.534	28.983
PS:Offset	1.1313	3.6684	3.4016	1.8151	2.4014	2.5233	5.6360	**0.2192**	**3.4823**	0.2380	3.2328	0.2254
SSIM	0.361	0.837	0.878	0.879	0.873	0.872	0.883	0.923	0.917	0.911	0.886	0.901
SS:Offset	0.0724	0.0571	0.0412	0.0142	0.0432	0.0312	0.0289	0.0004	0.0381	0.0005	0.0352	0.0007
Unconstrained DL Case 6: Exposure 5:3												
MAE	1.7010	2.0560	2.2449	2.2331	2.1969	2.3640	2.2324	**0.9266**	0.9322	2.1731	0.9281	0.9279
PSNR	14.515	25.278	25.702	25.705	25.802	25.536	25.504	**29.983**	**28.775**	28.745	26.356	27.651
PS:Offset	1.108	2.099	8.5371	7.1472	6.9022	6.2460	6.5837	**0.7210**	**8.134**	0.9730	7.8622	0.9274
SSIM	0.37	0.828	0.841	0.843	0.841	0.840	0.844	0.921	0.905	0.887	0.855	0.861
SS:Offset	0.073	0.0220	0.1060	0.0928	0.0932	0.0924	0.0941	0.0042	0.0595	0.0073	0.0912	0.0071
Time (S)	2160	1630	0.507	0.37	0.39	0.524	0.31	0.34/scale	0.34/scale	0.34/scale	0.33/scale	0.31/scale
Size (M)	-	-	8.06	4.17	5.09	21.69	4.83	5.98	5.98	5.98	5.90	5.93

TABLE S2: Data distribution

Type	Case 1 (1:3)	Case 2 (4:3)	Case 3 (3:5)	Case 4 (3:1)	Case5 (3:4)	Case 6 (5:3)	Case 7 (1:1)	Total DL-images
Training	4,159	4,403	4,600	4,159	4,403	4,600	17,319	43,643
Testing	837	803	805	837	803	805	3,318	8,208

method is implemented using `Pytorch 1.1.0` in a server with Intel Xeon processor and an Nvidia RTX 2080 TI GPU. For training our model, we use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and set the batch-size as four. Following [22], [15], we consider 256×256 patches for training. To aid generalization we perform random chromatic transformation (brightness, contrast and saturation sampled uniformly from [0.15,0.85]) and Gaussian noise-addition ($\sigma = 0.01$). The decimation step-size in our adaptive scale-space approach is selected as $\frac{1}{\sqrt{2}}$ (following traditional scale-space deblurring methods [5], [17]). The learning rate is decayed from 0.001 with a power of 0.3, and convergence is observed within 4,00,000 iterations.

We next analyse the computational cost of various DL-BMD methods (Table S1). As compared to the model-based optimization methods [5], [18], deep learning based methods are very efficient in terms of processing time. In particular, our method is atleast two orders faster than the competing techniques. In terms of memory requirement, the standard scale-space approach [15] is quite expensive due to the usage of independent network weights over three network-levels. Also, this memory requirement further increases if we have to increase the network-levels [15]. In contrast, our transformation allows the same network to be used for multiple-scales, thereby maintaining the *same* memory requirement for diverse network-levels. Overall, our deblurring network's memory requirement is comparable to the competing state-of-the-art deep learning methods [15], [21], [3], [21], [22] while exhibiting state-of-the-art performance.

(c) **Dataset Generation**: Since unconstrained dynamic scene blur has not been hitherto addressed, we created a dataset with diverse exposure (like the constrained case in [22]). We follow a similar procedure as that of [22] in creating DL blur dataset. We next highlight our differences. As typically followed in single-lens dynamic scene blur generation [6], a blurry image is generated by averaging a sharp high frame rate sequence to approximate a long exposure. The dataset in [22] consists of a wide variety of scenarios, both indoor and outdoor, which include diverse illumination, weather, and motion patterns. To increase the video frame rate, it employs a fast and high-quality frame interpolation method [7] and generate different blur-sizes by: (a) considering equal number of synchronized set of DL image-pairs in both left- and right-views (i.e., identical and fully-overlapping exposures) (b) ground truth frame is temporally centered on the exposure time. The major difference of our data generation is regarding the points (a) and (b). For Point (a), following [5], we do *not* consider synchronization between DL image-pairs, but allow a random non-zero exposure-intersection (as shown in Fig. 2(a)). Further we allow different number of DL image-pairs in the two-views in the ratio $1:1$, $1:3$, $3:1$, $3:4$, $4:3$, $3:5$, and $5:3$ (totalling seven exposure-cases). For all cases, following [5], exposure-overlap is randomly sampled from 10-100% with standard resolution 1:2. Table S2 provides the distribution of data samples. For Point (b), we consider the ground truth frame to be temporally

centered on the intersection of the left-right view exposure times to ensure ground truth is view-consistent (unlike the case of [22], as illustrated in Fig. 2(a)). (We will release our dataset upon acceptance.)

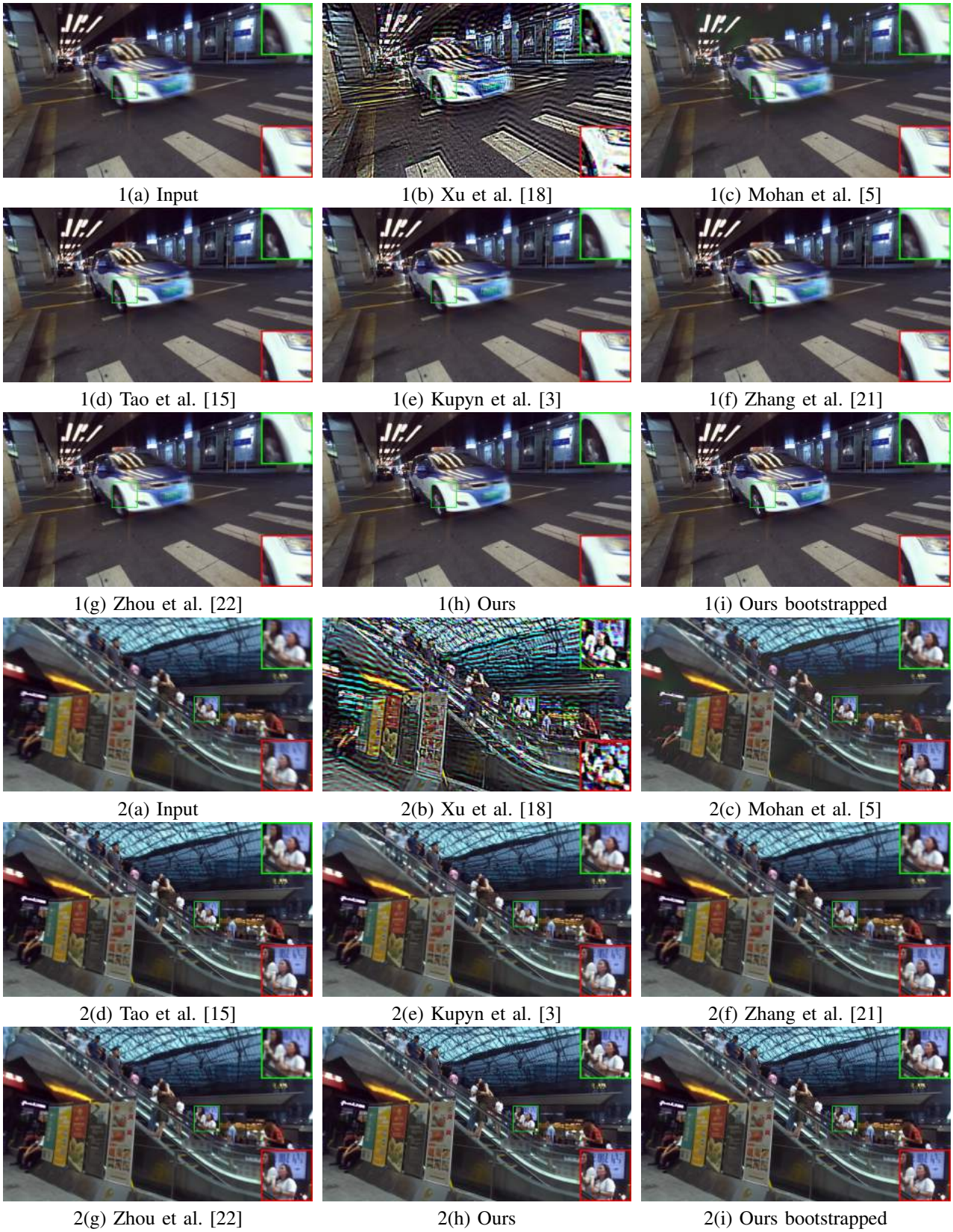
(d) We provide in Figures S3–S4 a detailed comparisons of the examples in the main paper (Figs. 9–10). Further, Table S1 provides the evaluation on remaining cases of Table III, and Figs. S1(b-c) provide the SSIM evaluation of Sec. V-B1.

(e) Figures S5–S6 provide qualitative results of the implication of view-consistency for DL super-resolution [16] (Sec. V-B3). (Note that for quantitative evaluation in Fig. 8(d), as ground truth super-resolved results are not available, we used super-resolved DL clean images as the reference for all methods.)

(f-h) **Additional Evaluations**: We provide more results on unconstrained DL static scene blur examples (from [5]) and constrained DL blur examples (from [22]) in Figs. S10-S12 and Fig. S13, respectively. Further, we provide left-right views of three unconstrained examples from our dataset in Figs. S7-S9. A thorough evaluation reveals that our method is comparable with state-of-the-art methods. Further, we include ten more examples with detailed comparisons in Figs. S14-S18, covering diverse unconstrained DL dynamic scene blur cases. In all the cases, our method produces view-consistent results and performs better than the state-of-the-art methods.

REFERENCES

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.
[2] H. Gao, X. Tao, X. Shen, and J. Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, pages 3848–3856, 2019.
[3] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*, pages 8183–8192. IEEE, 2018.
[4] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*. IEEE, October 2019.
[5] M. R. M. Mohan, S. Girish, and A. N. Rajagopalan. Unconstrained motion deblurring for dual-lens cameras. In *ICCV*. IEEE, October 2019.
[6] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017.
[7] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pages 261–270. IEEE, 2017.
[8] A. Oppenheim and R. Schafer. *Discrete Time Signal Processing*. Prentice-Hall, 2014.
[9] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636. IEEE, 2016.
[10] L. Pan, Y. Dai, M. Liu, and F. Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *CVPR*, pages 6987–6996. IEEE, 2017.
[11] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, pages 2909–2916. IEEE, 2014.
[12] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman. Deep generative filter for motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2993–3000, 2017.
[13] A. Sellent, C. Rother, and S. Roth. Stereo video deblurring. In *ECCV*, pages 558–575. Springer, 2016.
[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representation (ICLR)*, pages 1–8. IEEE, 2015.
[15] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018.
[16] L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo. Learning parallax attention for stereo image super-resolution. In *CVPR*, pages 12250–12259. IEEE, 2019.
[17] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012.
[18] L. Xu and J. Jia. Depth-aware motion deblurring. In *IEEE International Conference on Computational Photography*, pages 1–8. IEEE, 2012.
[19] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 1790–1798, 2014.
[20] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114. IEEE, 2013.
[21] H. Zhang, Y. Dai, H. Li, and P. Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*. IEEE, June 2019.
[22] S. Zhou, J. Zhang, W. Zuo, H. Xie, J. Pan, and J. S. Ren. Davanet: Stereo deblurring with view aggregation. In *CVPR*, pages 10996–11005. IEEE, 2019.



1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S3: Detailed comparisons of Figs. 9(1,2) in the main paper.

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

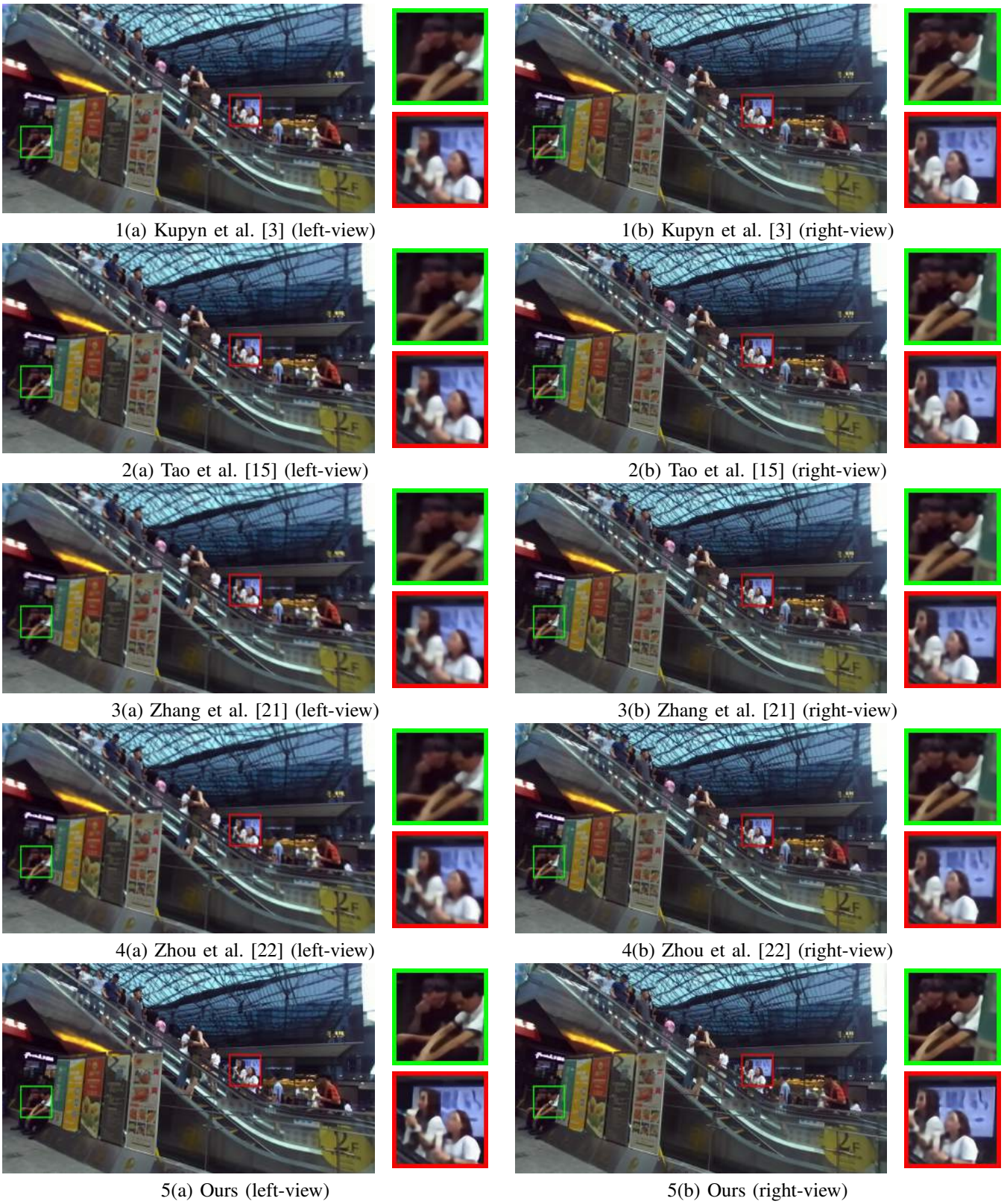
2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

Fig. S4: Detailed comparisons of Fig. 9(3) and Fig. 10(1) in the main paper.



1(a) Kupyn et al. [3] (left-view) 1(b) Kupyn et al. [3] (right-view)

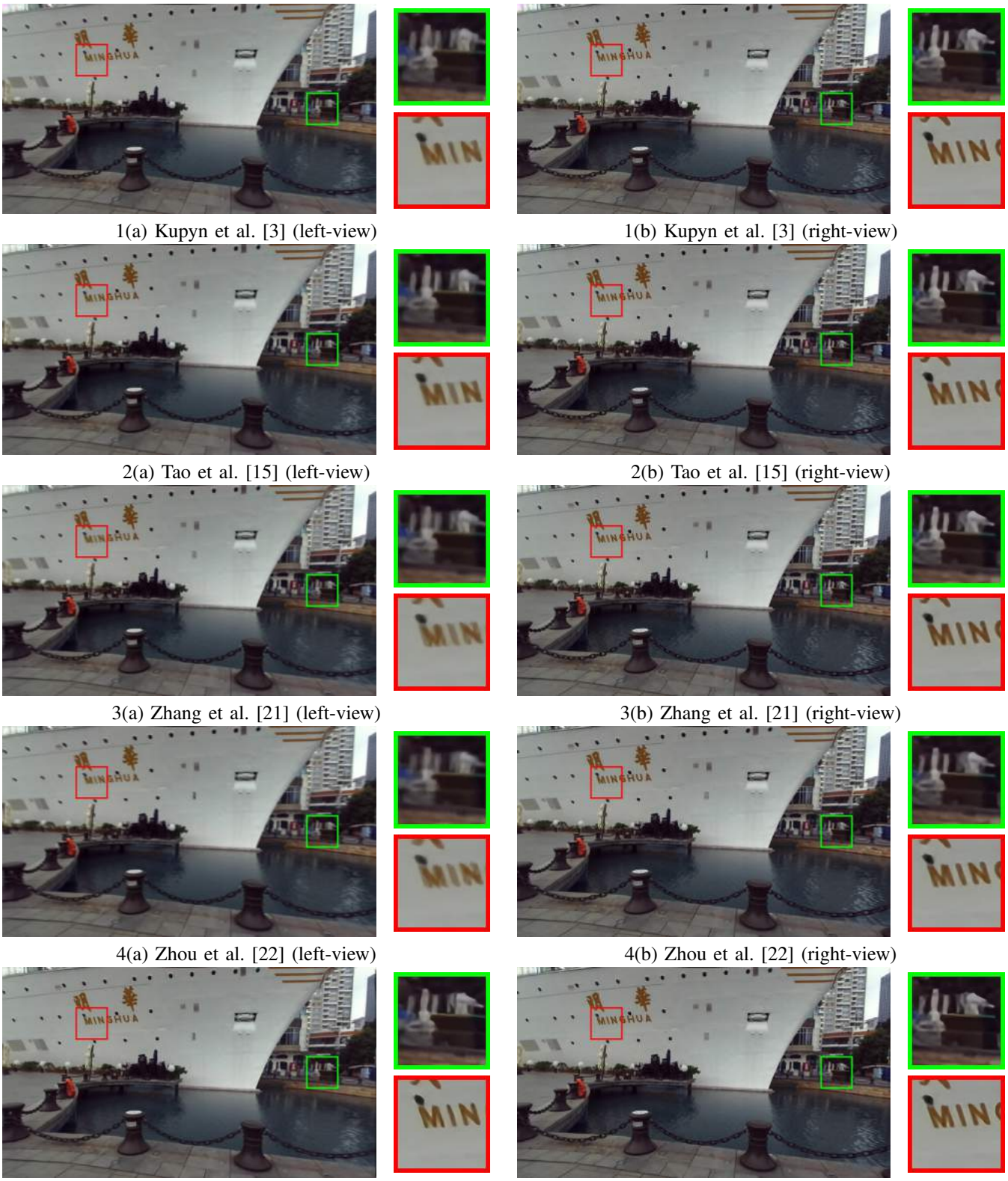
2(a) Tao et al. [15] (left-view) 2(b) Tao et al. [15] (right-view)

3(a) Zhang et al. [21] (left-view) 3(b) Zhang et al. [21] (right-view)

4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S5: Example 1: Applicability of different deblurring methods for DL super-resolution [16]. Note that our method is able to produce the desired view-consistent super-resolution results.



1(a) Kupyn et al. [3] (left-view) 1(b) Kupyn et al. [3] (right-view)

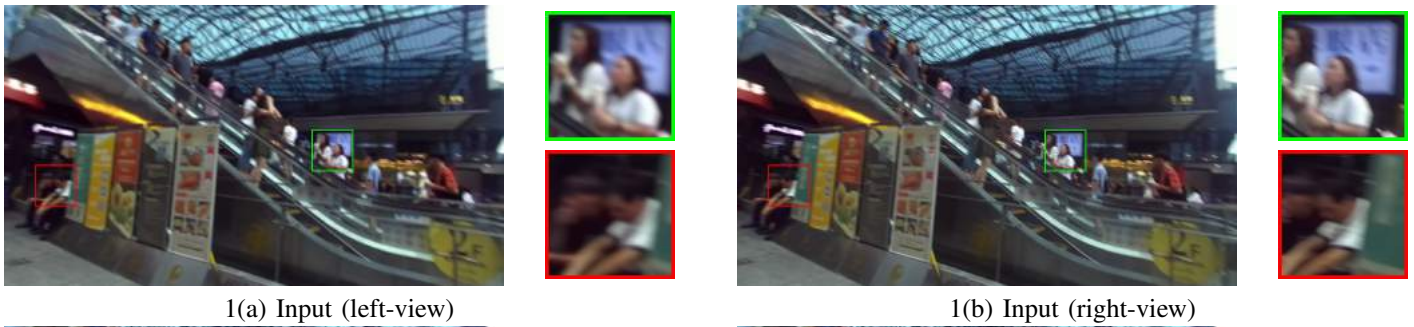
2(a) Tao et al. [15] (left-view) 2(b) Tao et al. [15] (right-view)

3(a) Zhang et al. [21] (left-view) 3(b) Zhang et al. [21] (right-view)

4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S6: Example 2: Applicability of different deblurring methods for DL super-resolution [16]. Note that our method is able to produce the desired view-consistent super-resolution results.



1(a) Input (left-view) 1(b) Input (right-view)

2(a) Ramakrishnan et al. [12] (left-view) 2(b) Ramakrishnan et al. [12] (right-view)

3(a) Kupyn et al. [3] (left-view) 3(b) Kupyn et al. [3] (right-view)

4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S7: Example 1: Left-right view comparisons in our unconstrained DL dataset. Our method is able to enhance the deblurring performance in the left-view (which undergoes more degradations) and produce view-consistent results as compared to the competing methods.

1(a) Input (left-view) 1(b) Input (right-view)

2(a) Ramakrishnan et al. [12] (left-view) 2(b) Ramakrishnan et al. [12] (right-view)

3(a) Kupyn et al. [3] (left-view) 3(b) Kupyn et al. [3] (right-view)

4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S8: Example 2: Left-right view comparisons in our unconstrained DL dataset. Our method is able to enhance the deblurring performance in the left-view (which undergoes more degradations) and produce view-consistent results as compared to the competing methods.

1(a) Input (left-view) 1(b) Input (right-view)

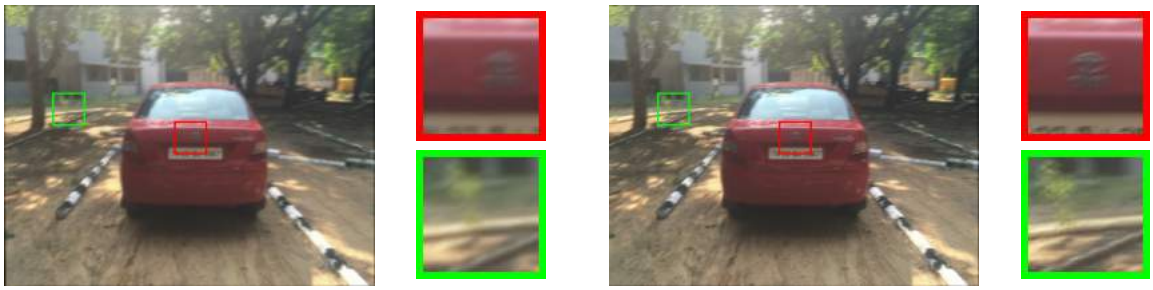
2(a) Ramakrishnan et al. [12] (left-view) 2(b) Ramakrishnan et al. [12] (right-view)

3(a) Kupyn et al. [3] (left-view) 3(b) Kupyn et al. [3] (right-view)

4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

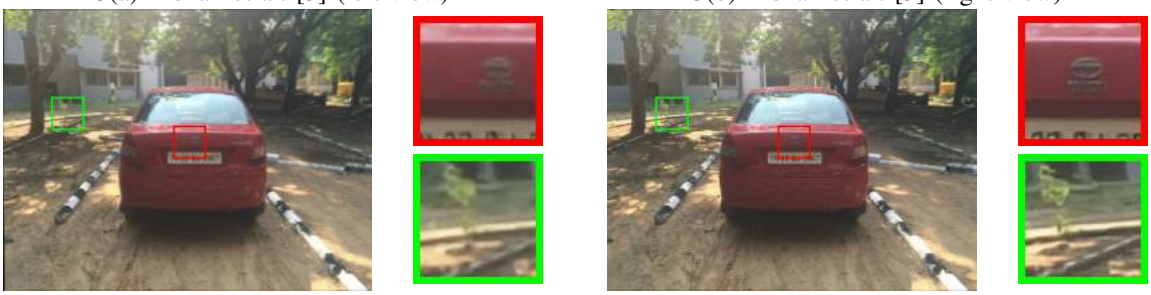
Fig. S9: Example 3: Left-right view comparisons in our unconstrained DL dataset. Our method is able to enhance the deblurring performance in the left-view (which undergoes more degradations) and produce view-consistent results as compared to the competing methods.



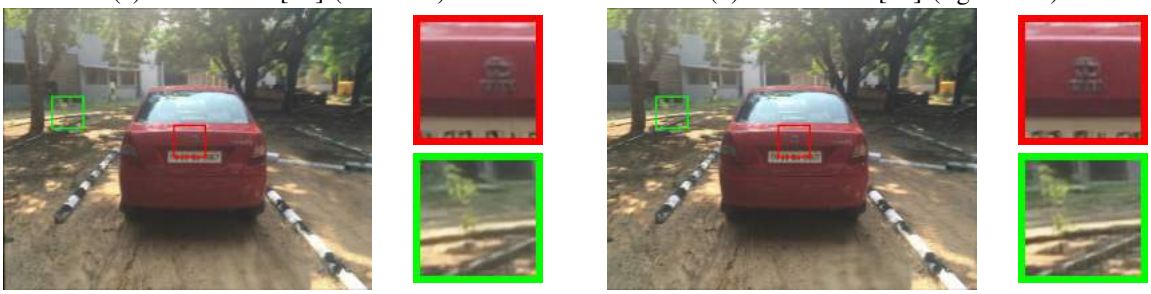
1(a) Input (left-view) 1(b) Input (right-view)

2(a) Tao et al. [15] (left-view) 2(b) Tao et al. [15] (right-view)

3(a) Mohan et al. [5] (left-view) 3(b) Mohan et al. [5] (right-view)

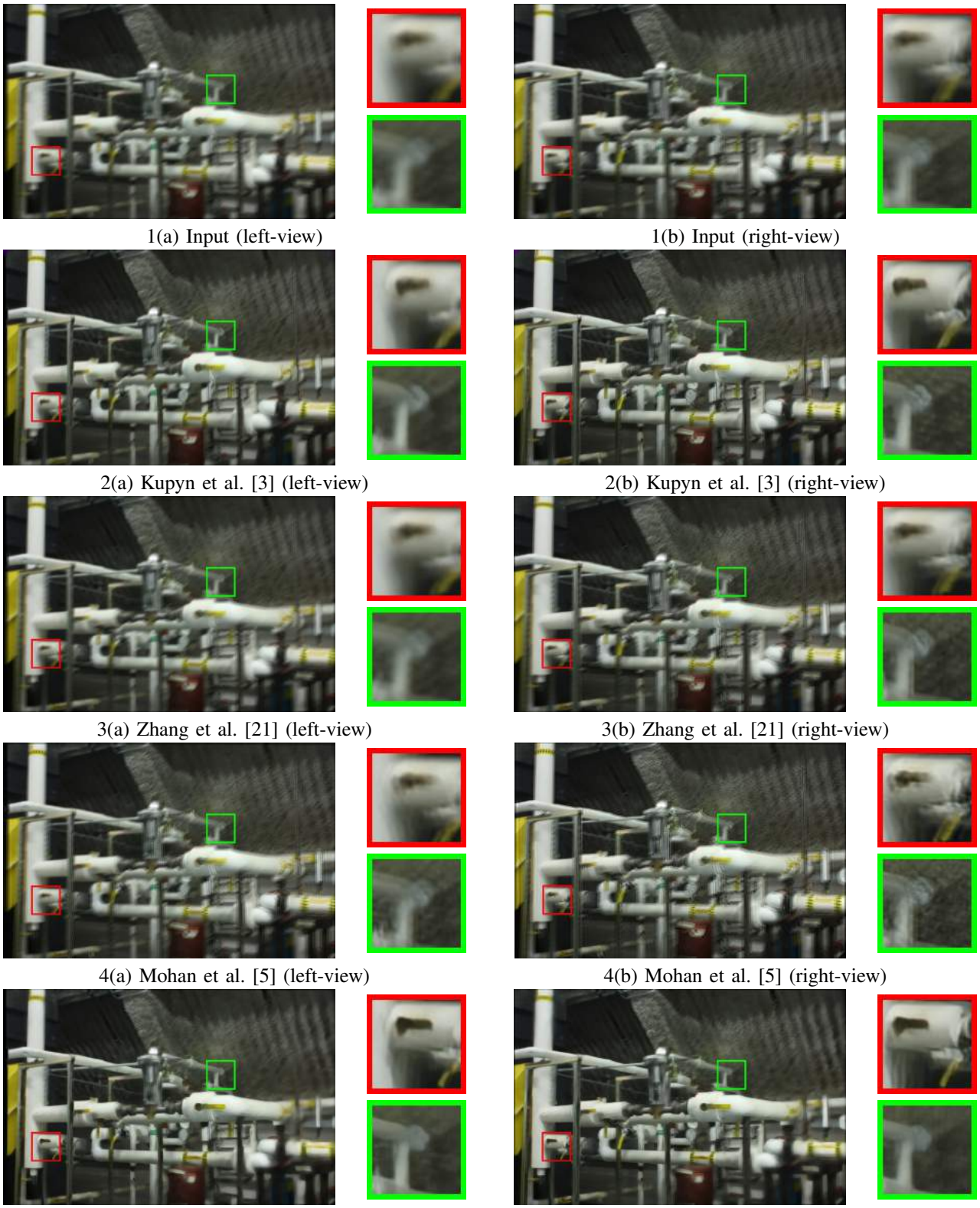


4(a) Zhou et al. [22] (left-view) 4(b) Zhou et al. [22] (right-view)



5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S10: Example 1: Comparisons for unconstrained static scene blur case (from [5]). Our method is able to produce view-consistent results as compared to the competing methods. Note that the computational cost of [5] is very high as compared to our method (Table. S1).



1(a) Input (left-view) 1(b) Input (right-view)

2(a) Kupyn et al. [3] (left-view) 2(b) Kupyn et al. [3] (right-view)

3(a) Zhang et al. [21] (left-view) 3(b) Zhang et al. [21] (right-view)

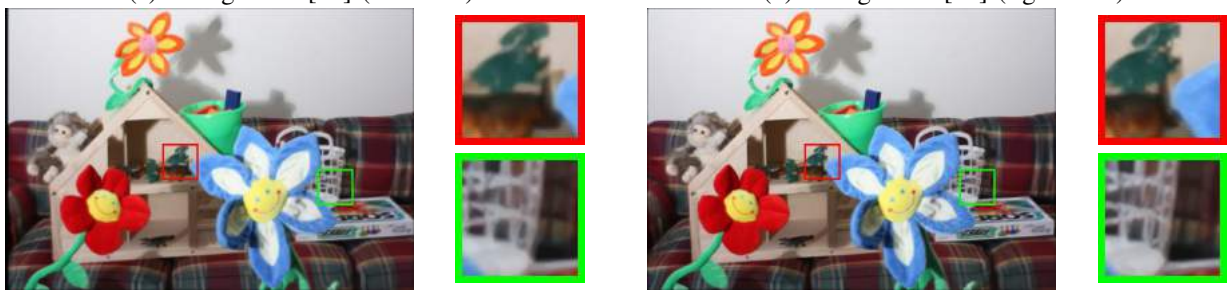
4(a) Mohan et al. [5] (left-view) 4(b) Mohan et al. [5] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

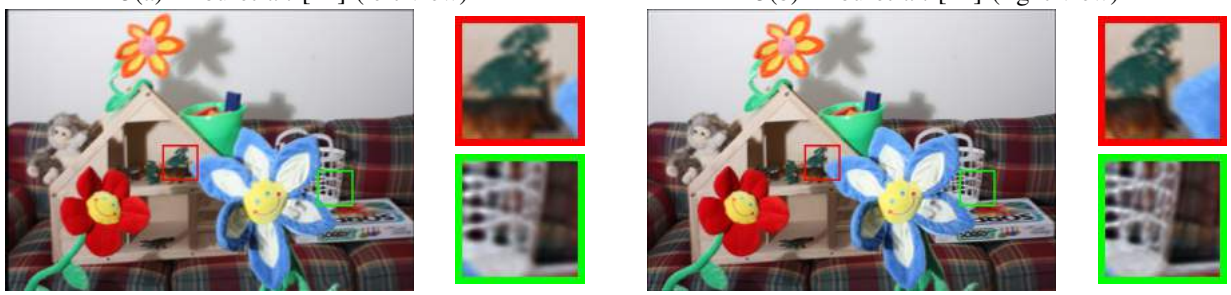
Fig. S11: Example 2: Comparisons for unconstrained static scene blur case (from [5]). Our method is able to produce view-consistent results as compared to the competing methods. Note that the computational cost of [5] is very high as compared to our method (Table. S1).

1(a) Input (left-view) 1(b) Input (right-view)

2(a) Zhang et al. [21] (left-view) 2(b) Zhang et al. [21] (right-view)



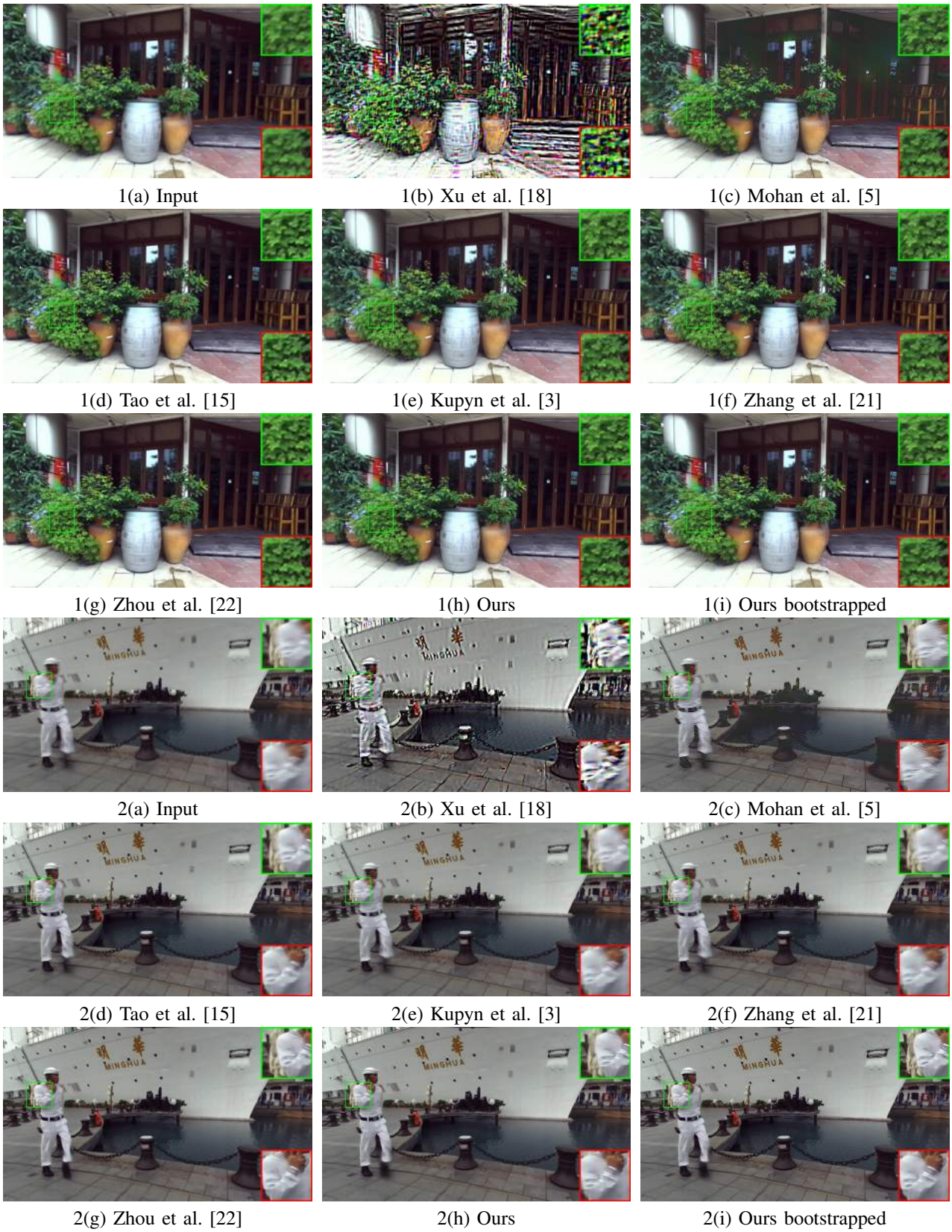
3(a) Zhou et al. [22] (left-view) 3(b) Zhou et al. [22] (right-view)



4(a) Mohan et al. [5] (left-view) 4(b) Mohan et al. [5] (right-view)

5(a) Ours (left-view) 5(b) Ours (right-view)

Fig. S12: Example 3: Comparisons for unconstrained static scene blur case (from [5]). Our method is able to produce view-consistent results as compared to the competing methods. Note that the computational cost of [5] is very high as compared to our method (Table. S1).



1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

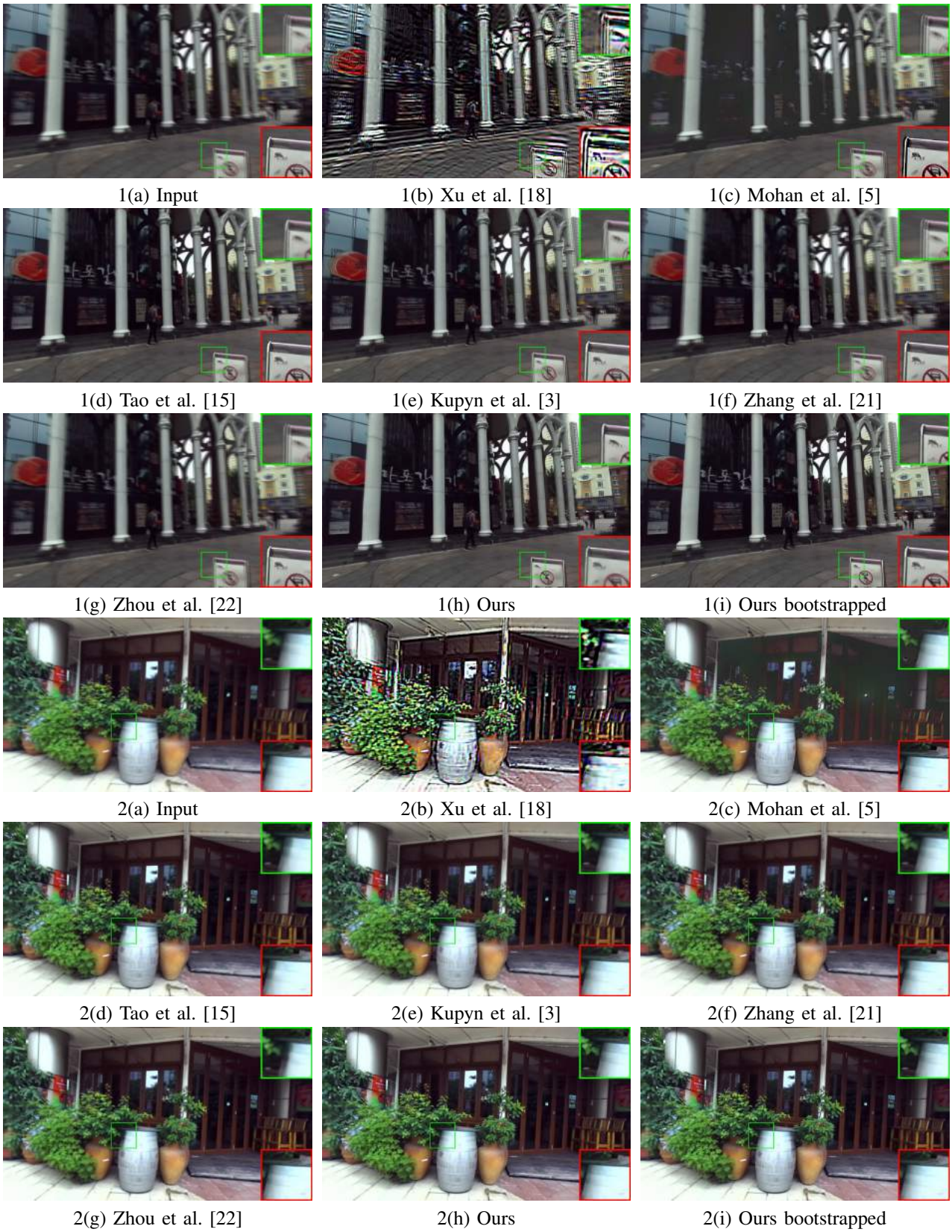
1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S13: Examples 1-2: Comparisons for constrained blur case (from [22]). Our method is able to produce view-consistent results as compared to the competing methods.



1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S14: Examples 1-2: Comparisons for unconstrained dynamic scene blur case (i.e., exposure 3:1 and 1:1). Our method is able to produce view-consistent results as compared to the competing methods. After bootstrapping, our method produces good view-*in*consistent result (please compare *both* views).

1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S15: Examples 3-4: Comparisons for unconstrained dynamic scene blur case (i.e., exposure 3:1 and 4:1). Our method is able to produce view-consistent results as compared to the competing methods. After bootstrapping, our method produces good view-*in*consistent result (please compare *both* views).

1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

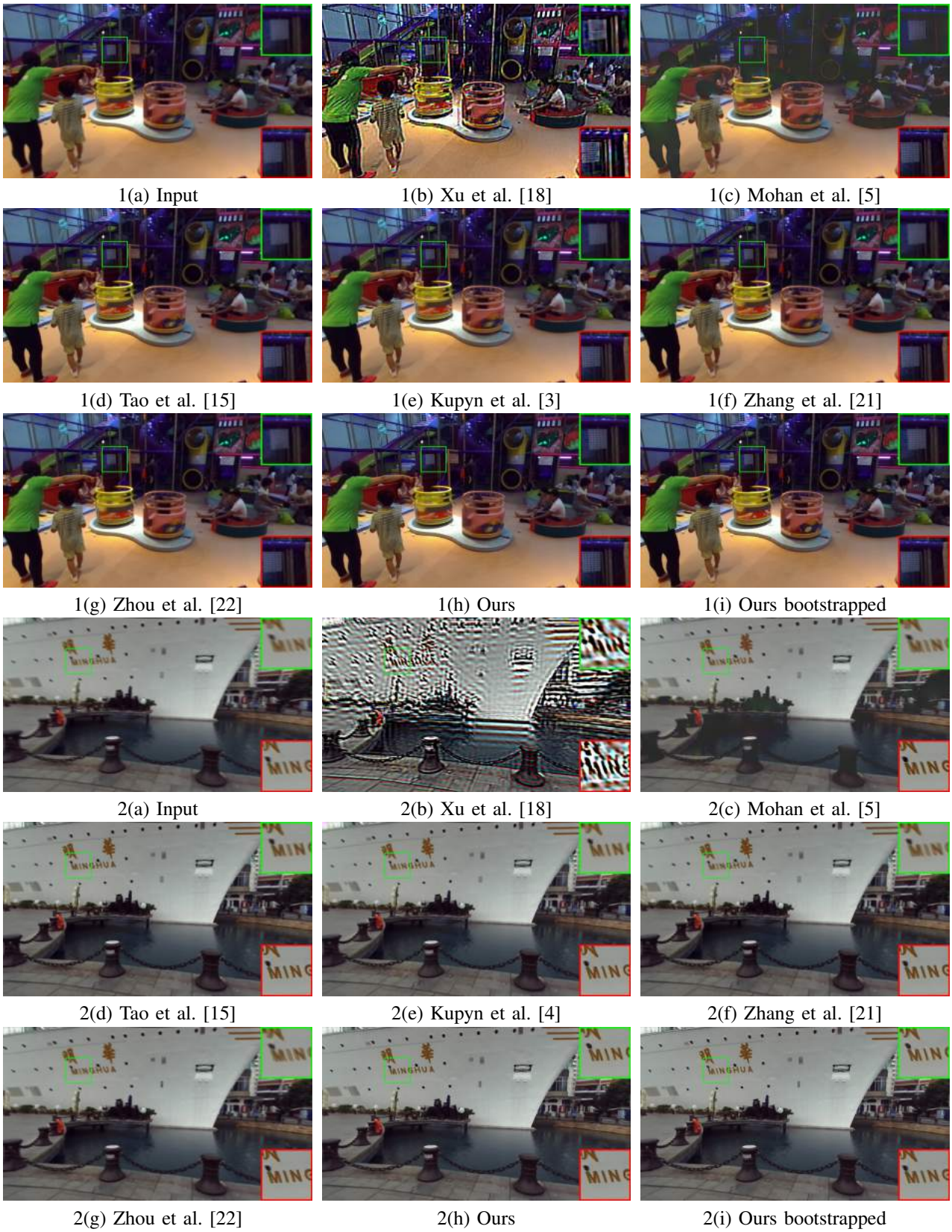
1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S16: Examples 5-6: Comparisons for unconstrained dynamic scene blur case (i.e., exposure 3:4 and 4:3). Our method is able to produce view-consistent results as compared to the competing methods. After bootstrapping, our method produces good view-*in*consistent result (please compare *both* views).



1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [4] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S17: Examples 7-8: Comparisons for unconstrained dynamic scene blur case (i.e., exposure 3:4 and 4:3). Our method is able to produce view-consistent results as compared to the competing methods. After bootstrapping, our method produces good view-*in*consistent result (please compare *both* views).

1(a) Input 1(b) Xu et al. [18] 1(c) Mohan et al. [5]

1(d) Tao et al. [15] 1(e) Kupyn et al. [3] 1(f) Zhang et al. [21]

1(g) Zhou et al. [22] 1(h) Ours 1(i) Ours bootstrapped

2(a) Input 2(b) Xu et al. [18] 2(c) Mohan et al. [5]

2(d) Tao et al. [15] 2(e) Kupyn et al. [3] 2(f) Zhang et al. [21]

2(g) Zhou et al. [22] 2(h) Ours 2(i) Ours bootstrapped

Fig. S18: Examples 9-10: Comparisons for unconstrained dynamic scene blur case (i.e., exposure 5:3 and 3:5). Our method is able to produce view-consistent results as compared to the competing methods. After bootstrapping, our method produces good view-*in*consistent result (please compare *both* views).